VŠB– Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra telekomunikační techniky

Monitorování síťových zařízení pomocí NetFlow

Network Monitoring with NetFlow

2017

Bc. Lukáš Fojtík

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra telekomunikační techniky

Zadání diplomové práce

Student: **Bc. Lukáš Fojtík**

Studijní program: N2647 Informační a komunikační technologie

Studijní obor: 2612T059 Mobilní technologie

Téma: Monitorování síťových zařízení pomocí NetFlow
Network Monitoring with NetFlow

Jazyk vypracování: čeština

Zásady pro vypracování:

Monitorování sítí patří k důležitým činnostem při správě sítí. Cílem diplomové práce je navrhnout řešení pro monitorování síťového provozu pomocí nástroje NetFlow, a jeho následné vykreslování pomocí nástroje RRDTool.

Řešení práce musí splňovat následující body:
1. Studium a popis protokolu NetFlow.
2. Instalace a konfigurace síťových prvků.
3. Testování a vyhodnocování síťového provozu v laboratorním prostředí.
4. Vykreslování grafů pomocí RRDTool.

Seznam doporučené odborné literatury:

[1]LUCAS, W., Michael. *Network Flow Analysis.* No Starch Press; 1 edition 2010. ISBN-13: 978-1593272036
[2]SANTOS, Omar. *Network Security with NetFlow and IPFIX: Big Data Analytics for Information Security.* Cisco Press; 1 edition 2015. ISBN-13: 978-1587144387

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **Ing. Pavel Nevlud**

Datum zadání: 01.09.2016
Datum odevzdání: 28.04.2017

doc. Ing. Miroslav Vozňák, Ph.D.
vedoucí katedry

prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlášení studenta

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně.Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě dne: *26. dubna 2017*

.....
podpis studenta

Poděkování

Děkuji vedoucímu práce Ing. Pavlu Nevludovi za poskytnutou pomoc a cenné informace při zpracování diplomové práce.

Abstrakt

Cílem této diplomové práce je navrhnout řešení pro monitorování síťového provozu pomocí NetFlow, a jeho následné vykreslování pomocí nástroje RRDTool.

V první části je popsán samotný protokol NetFlow, konkrétně jeho způsob fungování, verze a topologie. Rovněž jsou zde zmíněny NetFlow analyzátory a nástroj pro vykreslování grafů - RRDTool. V dalších částech jsou uvedeny podrobně konfigurace NetFlow na počítačích s operačním systémem Linux a na Cisko směrovačích. Jsou zde také popsány konfigurace generátorů provozu, databáze pro ukládání hodnot a skriptu, který automaticky zpracovává nasbírané NetFlow data, z nichž pak vytváří grafy v programu RRDTool.

Klíčová slova

NetFlow, Flexibilní NetFlow, Nfdump, Nfcapd, datový tok, kolektor, exportér, graf, Mgen, Mausezahn, RRDTool, Cisco, Linux

Abstract

The aim of this thesis is to propose a solution for monitoring network traffic using NetFlow, and its subsequent rendering using RRDTool.

The first part describes NetFlow protocol itself, specifically its mode of operation, versions and topologies. There are also mentioned NetFlow analyzers and the tool for drawing graphs - RRDTool. In other parts are detailed NetFlow configurations on computers running Linux and Cisko routers. There are also described the configurations of packet generators, a database for storing values and a script that automatically processes the collected NetFlow data from which then creates graphs in the program RRDTool.

Keywords

NetFlow, Flexible NetFlow, Nfdump, Nfcapd, flow, collector, exporter, graph, MGEN, Mausezahn, RRDTool, Cisco, Linux

Obsah

Seznam použitých symbolů a zkratek

Zkratka	Anglický význam	Český význam
BGP	Border Gateway Protocol	Dynamický směrovací protokol
FIN	-	Příznak pro ukončení spojení
ICMP	Internet Control Message Protocol	Protokol pro generování kontrolních zpráv
IOS	Internetwork Operating System	Operační systém na Cisco zařízeních
IP	Internet Protocol	Internetový protokol
IPFIX	IP Flow Information Export	Standart pro export NetFlow dat
IPv4	Internet Protocol version 4	Internetový protokol verze 4
IPv6	Internet Protocol version 6	Internet Protocol version 6
ISP	Internet Service Provider	Poskytovatel internetového přIPojení
kB	kilobytes	Jednotka počtu přenesených dat
MPLS	MultIProtocol Label Switching	Protokol pro přenos paketů pomocí značek
MRTG	Muti Router Traffic Grapher	Nástroj pro vytváření grafů
Mz	MauseZahn	Nástroj pro generování provozu
OSPF	Open Shortest Path First	Interní směrovací protokol
OSPFv3	Open Shortest Path First version 3	Interní směrovací protokol verze 3
PC	Personal Computer	Osobní počítač
pr	Protocol	Protokol
RRDTool	Round-Robin Database Tool	Databáze
RTP	Realtime Transport Protocol	Protokol pro přenos audio a video paketů
SYN	-	Příznak pro navázání spojení
TCP	Transmission Control Protocol	Spojově orientovaný protokol
UDP	User Datagram Protocol	Nespojově orientovaný protokol

Seznam ilustrací

Úvod

Protokol NetFlow byl vyvinutý společností Cisco Systems. Nejčastěji se využívá pro monitorování vstupního provozu u ISP providerů, v podnikových sítích, sledování toků mezi BGP autonomními systémy a v neposlední řadě jej využívají poskytovatelé IP služeb, kteří podle mohou zákazníkům účtovat přenášená data

Tato práce se zabývá způsoby a možnostmi monitorování sítě pomocí nástroje NetFlow. V první části práce je popis protokolu NetFlow, jeho funkčnosti a schémat zapojení. Kromě protokolu NetFlow je zde popsán i jeho nástupce na Cisco zařízeních - Flexibilní NetFlow. Kromě toho je zde stručně popsán nástroj RRDTool a různé open-source NetFlow analyzátory, především analyzátor Softflowd a Nfdump.

V praktické části je popsána konfigurace NetFlow na počítačích s operačním systémem Linux, kde jeden počítač analyzuje provoz na jeho síťovém rozhraní. Tento analyzovaný provoz je poslán na druhý počítač, který tyto data zpracovává a vytváří z nich grafy pomocí nástroje RRDTool.

Dále je zde také popis konfigurace NetFlow na Cisco zařízeních (směrovačích) Jedná se o analýzu síťového provozu na Cisco směrovačích a jejich následné zasílání na počítač s operačním systémem Linux k dalšímu zpracování a vykreslení výsledných grafů pomocí nástroje RRDTool.

Rovněž je zde detailně popsána konfigurace Flexibilního NetFlow, včetně výpisů důležitých tabulek na jednotlivých počítačích a směrovači.

1. Protokol NetFlow

1.1. Úvod do NetFlow

Protokol NetFlow byl původně určen jako doplňková služba k Cisco směrovačům. Poskytuje (i v reálném čase) data jako jsou zdrojové a cílové IP adresy, zdrojové a cílové porty, počty přenesených paketů, bajtů a další. Po nakonfigurování funkce NetFlow na směrovači je možno nasbírané toky přímo prohlížet nebo se mohou posílat na zařízení v síti, kde je možno NetFlow ukládat a provádět podrobnější analýzy[1].

1.2. PrincIP NetFlow

NetFlow je založen na princIPu toků síťového provozu. Za IP tok se považuje úplná síťová konverzace. Například navázání TCP spojení z pracovní stanice na www server při čtení webové stránky je jeden tok. Začátek TCP spojení představuje začátek toku a uzavřením spojení dochází k ukončení IP toku. Dalším příkladem toku je posloupnost ICMP paketů při testování pomocí pingu. Tok začíná prvním ICMP paketem a končí posledním. Každý tok je popsán unikátní skupinou následujících údajů:

- zdrojová IP adresa
- cílová IP adresa
- zdrojový port
- cílový port
- číslo IP protokolu
- typ služby
- Vstupní rozhraní [1]

Tyto údaje jsou pro celý tok neměnné. Načtení více webových stránek přes jedno TCP spojení bude tedy jediný tok. Pokud však dojde k více spojením s jediným web serverem, tak každé spojení bude mít jiný zdrojový port a každé tak bude představovat samostatný tok. Například útočník skenující cílové IP adresy bude generovat mnoho toků (každý pro samostatnou dvojici zdrojového a cílového portu). Za velice důležité upozornění stojí fakt, že tok je jednosměrný. Jedna adresa je vždy pouze zdrojem provozu a druhá adresa je vždy cílem. To znamená, že tok je tvořen pouze provozem v jednom směru. Při komunikaci s webovým serverem, tak vždy existují dva toky, jeden je od stanice k serveru a druhý je od serveru ke stanici[1].

Ve starších verzích IOSu nebylo možno specifikovat zda se bude sledovat provoz do rozhraní nebo z rozhraní. Dnešní směrovače to již umožňují (IP flow ingress,[egress]). Pokud bude nakonfigurováno to, že směrovač sleduje provoz přicházející na rozhraní zvenčí. Jestliže zapneme NetFlow na rozhraní, kde je přIPojená pracovní stanice, bude stanice vždy zdrojem dat ve všech tocích. Pokud NetFlow zapneme na jiném rozhraní, přes něž přichází provoz pro danou stanici, bude v tocích tohoto rozhraní stanice vystupovat jako cíl. Není vždy snadné zjistit, čím je tvořena jedna relace a kdy tedy určitý tok skončí. U TCP je to jasné – relace začíná paketem SYN a končí paketem FIN. U protokolu UDP či ICMP je to obtížnější a směrovač používá různé heuristické postupy, jimiž o konci rozhoduje[1].

Protože směrovač má k dispozici jen určité množství paměti pro uchování dat o jednotlivých tocích, musí občas z paměti staré toky odstranit, aby bylo uvolněno místo pro nové toky. Je nutné mít na paměti, že pokud směrovač něco vyhodnotí jako jeden tok, nemusí jít ve skutečnosti o celou konverzaci. První část TCP spojení například může být zaznamenána jako jeden tok, další dva toky mohou být prostřední části komunikace a čtvrtý tok je její závěrečná část. Stejně tak je možné, že daná konverzace bude zaznamenána jako jediný tok. Záleží pouze na tom, zda směrovač bude potřebovat uvolnit své prostředky. Standardně se na Cisco směrovačích uzavírá tok z jednoho z následujících důvodů[1]:

- konec TCP spojení
- žádný provoz v rámci toku za posledních 15 sekund
- tok trvá déle jak 30 minut
- zaplnění tabulky toků [1]

1.2.1. Export NetFlow

NetFlow může shromážděna data exportovat, takže je lze přijímat na serveru a tam zpracovat dle potřeby. Přijímací počítač se nazývá flow kolektor. Data se exportují prostřednictvím UDP paketů na IP adresu a port, které lze na směrovači nastavit. Vzhledem k tomu, že pro export se využívá protokolu UDP může se stát, že dojde ke ztrátě exportovaných dat. U verze NetFlow 5 se zavádí sekvenční čísla, takže kolektor se o ztrátě může dozvědět, avšak neexistuje možnost ztracená data někde získat. Na Cisco směrovačích lze nastavit export pouze na jeden kolektor (jeden pro IP protokol a jeden pro IPv6 protokol), pokud vznikne potřeba mít data na více místech, tak předávání musí zajistit první kolektor [1].

1.3. Verze NetFlow

Existuje několik různých verzí mechanismu NetFlow. První byla verze 1, kde každý tok obsahuje následující informace: zdrojová IP adresa

- cílová IP adresa
- zdrojový port
- cílový port
- adresa následujícího směrovače
- číslo vstupního rozhraní
- číslo výstupního rozhraní
- počet odeslaných paketů
- počet odeslaných bajtů
- hodnota sysUpTime při začátku toku
- hodnota sysUpTime při konci toku
- číslo IP protokolu
- typ služby, ToS [1]

1.3.1. Krátké seznámení s dalšími verzemi:

- Verze 5: pokud jsou k dispozici poskytuje i údaj o AS z protokolu BGP a zavádí se sekvenční čísla, která slouží pro detekci ztracených paketů. Tato verze je nejpoužívanější.
- Verze 6: podpora pro tunelovaný provoz
- Verze 7: využívá se pouze v přepínačích Cisco Catalyst 5000, které musí být vybaveny kartou NetFlow Feature
- Verze 8: zavádí agregaci získaných dat, výrazně se zmenšuje objem dat exportovaných ze směrovačů. Po zvolení vhodného agregačního schématu bude směrovač seskupovat toky stejných skupin do jediného agregovaného toku, jehož data se pak exportují jako celek.

- Verze 9: je založena na tzv. šablonách. Někdy označován také jako „flexibilní NetFlow“, protože umožňuje flexibilně nastavit, jaké informace z provozu datové sítě budou sledovány. Podporuje IPv6 a další položky, které NetFlow v5 opomíjí.
- Verze 10, nebo též IPFIX, je aktuální, zatím nepříliš rozšířený standard obecně uznávaný IETF. IPFIX dovoluje rozšířit datové toky o další informace o síťovém provozu. IPFIX je budoucností monitorování datových toků. [1][2][20]

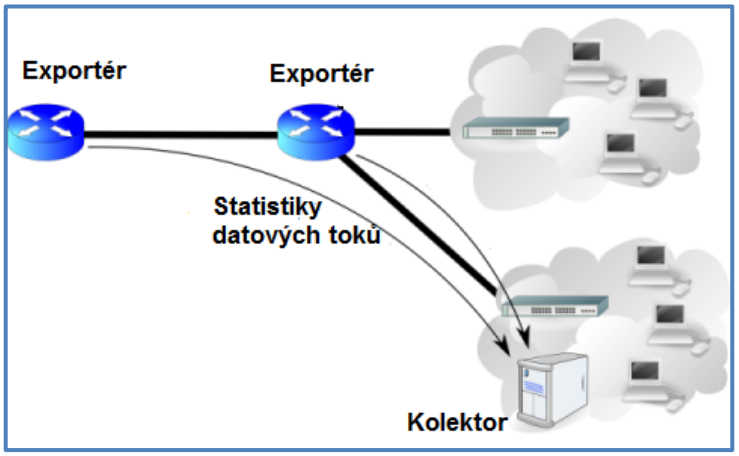
Nejrozšířenějšími verzemi NetFlow jsou verze 5 a 9. Potřeba jednotného a univerzálního standardu pro export informací o komunikaci v podobě IP toků dala vzniknout standardu IPFIX (Internet Protocol Flow Information Export). Ten definuje, jak jsou informace o IP tocích formátovány a přenášeny z exportéru (směrovače, switche, speciální sondy, ale i další zařízení) na kolektor. Dříve byli operátoři datových sítí odkázáni na proprietární standard NetFlow společnosti Cisco. Standard IPFIX lze označit za flexibilnějšího následníka NetFlow, který dovoluje rozšířit získávané datové toky o řadu důležitých informací o síťovém provozu. [1][20].

1.4. NetFlow architektura

NetFlow architektura se typicky skládá z několika NetFlow exportérů a jednoho NetFlow kolektoru.NetFlow exportér je připojen k monitorované lince a analyzuje procházející pakety. Na základě zachycených IP toků generuje NetFlow statistiky a ty exportuje na NetFlow kolektor. NetFlow kolektor je zařízení s velkou úložnou kapacitou, které sbírá statistiky z většího počtu NetFlow exportérů a ukládá je do dlouhodobé databáze. Nad těmito daty obvykle běží aplikace, která je umí efektivně vizualizovat a generovat z nich přehledy v podobě grafů a tabulek, které umožňují jednoduše analyzovat monitorovaný provoz i běžnému uživateli [3].

1.4.1. Tradiční architektura

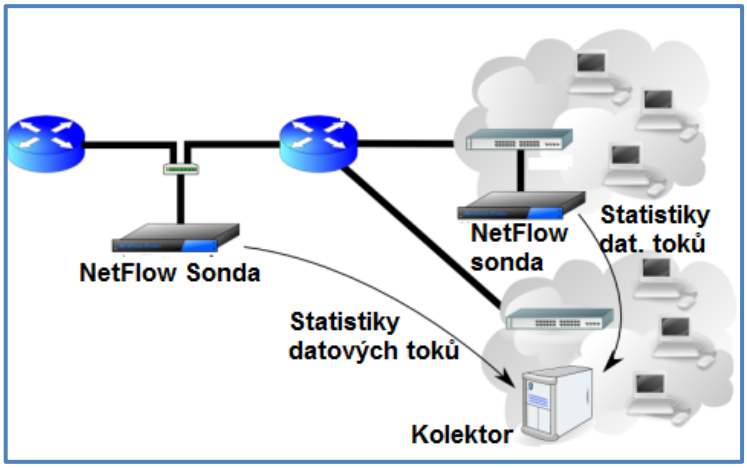
Tradiční architektura podle Cisco předpokládá na pozici NetFlow exportérů směrovače, které vedle své hlavní činnosti provádějí také výpočet NetFlow statistik. Tradiční architektura však trpí několika nevýhodami. Především se jedná o vysokou pořizovací cenu podobného zařízení, které komplikuje jeho nasazení v malých a středních sítích. Výpočet NetFlow statistik také omezuje směrovací výkon celého zařízení, proto většina směrovačů s podporou NetFlow (s výjimkou těch nejdražších) využívá na vstupu vzorkování, tzn. že se pro výpočet statistik využívá jen každý n-tý paket. Kromě snížené přesnosti měření omezuje vzorkování také pravděpodobnost odhalení bezpečnostních incidentů [3].



Obrázek 1.1: *Topologie tradiční architektury*

1.4.2. Moderní architektura

V poslední době se stávají velmi oblíbeným řešením využití pasivních NetFlow sond. NetFlow sondy odstraňují všechny nevýhody tradiční architektury a na rozdíl od směrovačů je lze přIPojit do libovolného bodu v síti a to transparentním způsobem. Sondy procházející data pouze monitorují a nijak do nich nezasahují (proto pasivní sondy). Exportované statistiky jsou na kolektor odesílány dedikovanou linkou a díky tomu jsou na monitorované lince zcela neviditelné. Tato vlastnost z nich činí velmi obtížný cíl pro případné útočníky [3].



Obrázek 1.2: *Topologie moderní architektury*

1.5. Flexibilní NetFlow

Jedná se o další generaci NetFlow. Hlavní výhodou této technologie je optimalizace sítě, snížení provozních nákladů, zlepšení plánování kapacity a odhalování bezpečnostních incidentů se zvýšenou flexibilitu a škálovatelnost. Schopnost charakterizovat provoz IP a určit jeho zdroj, dopravní cesty, jímž je načasování a informace aplikace je rozhodující pro dostupnosti sítě, výkon a řešení problémů. Sledování IP dopravních toků zvyšuje přesnost, kapacitní plánování a zajišťuje, že přidělování zdrojů podporuje organizačních cílů. Flexibilní NetFlow pomáhá zajistit, jak optimalizovat využití zdrojů, plánovat kapacitu sítě a identifikovat optimální aplikační vrstvy pro Quality of Service (QoS). [4]

1.5.1. Komponenty Flexibilní NetFlow

Flexibilní NetFlow se skládá z komponent, které mohou být použity v několika variantách a provádět analýzu provozu a export dat. Konkrétně se jedná o:

- Flow Monitory
- Flow Exportéry
- Záznamy (Records)
- Flow Samplery [5]

Každý Flow Monitor může obsahovat různé různé záznamy, flow exportéry a typ cache. Pokud se například změní IP adresa kolektoru v komponentě flow exportér dojte ke automatické změně na všech flow monitorech, u kterých je flow exportér aplikován.

Defaultně používá Flexibilní NetFlow verzi 9, jejíž hlavní vlastností je podpora IPv6 protokolu a exportování NetFlow dat ve formě šablon [5].

Záznam (record)

U Flexibilního NetFlow tvoří kombinace tzv. klíčových a neklíčových polí záznam. Tyto záznamy jsou určeny pro flow monitory. Klíčové pole v záznamu definují, na základě jakých kritérií se bude provoz sbírat, např. zdrojová IP adresa, cílová IP adresa, porty, atd. Neklíčové pole definuje, která část datového toku se bude sbírat a posílat na kolektor. Opět se může jednat o zdrojovou IP adresu, cílovou IP adresu, časové razítko startu datového toku, porty, počet bytů atd. Velkou výhodou Flexibilního NetFlow je, že uživatel si může nadefinovat podle jakých kritérií chce provoz sbírat a nebo může použít již existující šablonu, která sbírá datové toky podle originálního NetFlow [5].

Flow Monitor

Jedná se o další komponentu, která se aplikuje na rozhraní směrovače nebo přepínače. Skládá se z uživatelem předefinovaných záznamů, jednoho nebo více flow exportérů a cache, která je automaticky vytvořena po nastavení flow monitoru na síťové rozhraní. NetFlow data jsou sbírány ze síťového provozu a jsou ukládány do cache během monitorování na základě klíčových a neklíčových záznamů [5].

Flow exportér

Flow exporter posílá data z flow monitoru na vzdálený systém, typicky se jedná o NetFlow kolektor. Flow exportéry jsou vytvářeny jako oddělené entity při konfiguraci. Výhodou oproti originálnímu NetFlow je, že můžeme vytvořit několik flow exportérů, které se pak můžou přiřadit k jednomu nebo více flow monitorů. Také lze vytvořit jeden flow exportér a aplikovat ho na více flow monitorů [5].

Flow samplér

Hlavní úlohou flow sampléru je snížit zatížení, které vytváří Flexibilní NetFlow na síťovém zařízení. Pro to používá omezení počtu paketů, které jsou analyzovány. Lze nastavit rychlost vzorkování v určiném poměru. Například u poměru 1:2 se bude analyzovat každý druhý paket. Síťové zařízení tedy bude zpracovávat 50 % provozu [5].

1.5.2. Další výhody Flexibilního NetFlow

Mezi další výhody Flexible NetFlow patří:

- Flexibilita a škálovatelnost toku dat mimo tradiční NetFlow
- Přizpůsobené identifikace provozu
- Schopnost soustředit se a sledovat konkrétní chování sítě
- Schopnost sledovat širší škálu informací paketů a produkovat nové informace o chování sítě
- Možnost vytvoření několika Exportérů pro každý typ provozu, z každého exportéru lze posílat provoz různým kolektorům (originální NetFlow na Cisco posílal data maximálně na 2 kolektory)
- Podpora TCP a UDP portů
- Možnost analýzy CoS v paketech (originální NetFlow nepodporuje na Cisco)
- Možnost šifrování NetFlow dat [4]

1.6. NetFlow analyzátor Nfdump

Nfdump představuje sadu nástrojů, které shromažďují a zpracovávají NetFlow datové toky. Nfdump je založen na práci v příkazovém řádku, podobně jako Tcpdump. Podporuje NetFlow verze 1,5, 7, 9 a nejnovější verzi IPFIX. Díky tomu dokáže zpracovávat i IPv6 datové toky. Jeho další výhodou je flexibilní agregace - tedy z každého souboru se zachyceným provozem lze vybrat pouze ty data, které chceme [6].

1.6.1. Sada nástrojů programu Nfdump

Jak již bylo zmíněno, program Nfdump se skládá z více nástrojů, jejichž úkolem je zachycovat a různě zpracovávat NetFlow data:

Nfcapd

Hlavním úkolem nástroje Nfcapd je číst NetFlow data, přijaté ze sítě a ukládat je do souboru. Každých n minut (typicky 5 minut) dojde k vytvoření nového souboru s přijatými NetFlow daty. Tento soubor obsahuje v názvu datum a čas vytvoření [6].

Nfdump

Nfdump čte NetFlow data z uložených souborů, které vytvořil Nfcapd, které později může zobrazit na obrazovku, do CSV souboru apod. Rovněž může zpracovávat pouze ty data, které uživatel požaduje (například výpis všech zdrojových portů). Jeho syntaxe je podobná nástroji Tcpdump [6].

Nfprofile

Nfprofile čte NetFlow data ze souborů uložených Nfcapd. Podle nastavených pravidel dokáže tyto data filtrovat a ukládat pro pozdější využití [6].

Nfreplay

Nfreplay čte NetFlow data ze souborů uložených Nfcapd a odesílá je přes síť na jiného hostitele[6].

Nfclean.pl

Úkolem Nfclean.pl je vymazat staré soubory, které vytvořil Nfcapd. Tento skript může být spouštěn například každou hodinu [6].

1.7. NetFlow analyzátor SoftFlowd

SoftFlowd je síťový NetFlow analyzátor, který je schopný analzyovat a exportovat NetFlow data na příslušný kolektor. Může poslouchat síťové rozhraní nebo číst soubor se zachycenými datovými toky (například umí číst pcap soubory nebo soubory vytvořené programem Tcpdump). Tyto toky mohou být později odeslány na jiného hostitele (kolektor) nebo lokálně uloženy.

SoftFlowd podporuje NetFlow verze 1, 5, 9 a je plně kompatibilní s IPv6. Jakýkoliv standartní NetFlow kolektor by měl být schopen zpracovávat zprávy, přijaté od SoftFlowd.

Součástí SoftFlowd je i program SoftFlowctl, který umožňuje řízení běhu SoftFlowd a získávání statistických údajů o zachyceném provozu.

Tento nástroj byl vyvinut pro operační systémy Linux a OpenBSD [7][8].

1.8. Další Free Opensource NetFlow analyzátory

Níže je přehled několika nejpoužívanějších free open source NetFlow analyzátorů. Jejich největší výhodou je, že jejich užívání je bezplatné.

1.8.1. Ntop (Ntopng)

Pravděpodobně nejznámější open source síťový analyzér. Ntop funguje přes webové rozhraní a lze ho provozovat na operačních systémech Ubuntu x64, CentOS/Redhat x64 Linux distribucích, Windows x64, BeagleBoard ARM, Ubiquity sítě EdgeSměrovač a v neposlední řadě Mac OS X. Ntopng rovněž podporuje sFlow a IPFIX [9][10].

1.8.2. Flow-tools

Jedná se o knihovnu a soubor programů, které se používají ke sběru, posílání a generování reportů z NetFlow. Tento tool může být použit na jednom nebo i na více serverech. Floow-tools knihovna podporuje NetFlow ve verzích 1,5,7 [9][10].

1.8.3. FlowScan

FlowScan je více vizualizační tool, který analyzuje a reportuje NetFlow data. Rovněž může vytvářet různé grafy, které se podobají real-time grafům. FlowScan je vhodný pro operační systémy GNU/Linux nebo BSD systémy. Pro správnou funkci používá FlowScan následující softwarové balíky:

- cflowd - slouží jako kolektor
- flowscan - jedná se o skript v jazyce Perl, který propojuje ostatní softwarové balíky a je odpovědný za reporty
- RDDtool - tento tool ukládá nasbírané informace ve vlastní databázi [9][10].

1.8.4. EHNT

Neboli Extreme Happy NetFlow Tool. Tento tool provádí reporty v intervalech od 1 minuty do 24 hodin a poskytuje informace o IP protokolech, TCP/UDP portech atd. NetFlow tento tool podporuje ve verzi 5 [9][10].

1.8.5. BPFT

Neboli Berlekey Packet Filter Traffic collector. Používá se k zachycení IP provozu, skenování zdrojových / koncových IP adres a portů. Rovněž poskytuje zprávy o počtu odeslaných a přijatých bytů. Tyto informace ukládá do jednoho kompaktního binárního souboru [9][10].

1.8.6. Maji

Jedná se o implementaci IPFIX, která je založena na knihovně libtrace. IPFIX jako takové bylo vyvinuto pro sběr a exportování naměřených dat [9][10].

1.8.7. PmGraph

PmGraph je vynikající open source tool pro zobrazování a monitorování síťového provozu. K tomu používá pmacct (síťový monitorovací a auditovací tool, který monitoruje provoz pomocí NetFlow nebo Sflow na síťových zařízeních (switche, směrovačy)), nasbírané data později ukládá do databáze a analyzuje pomocí PmGraph [9][10].

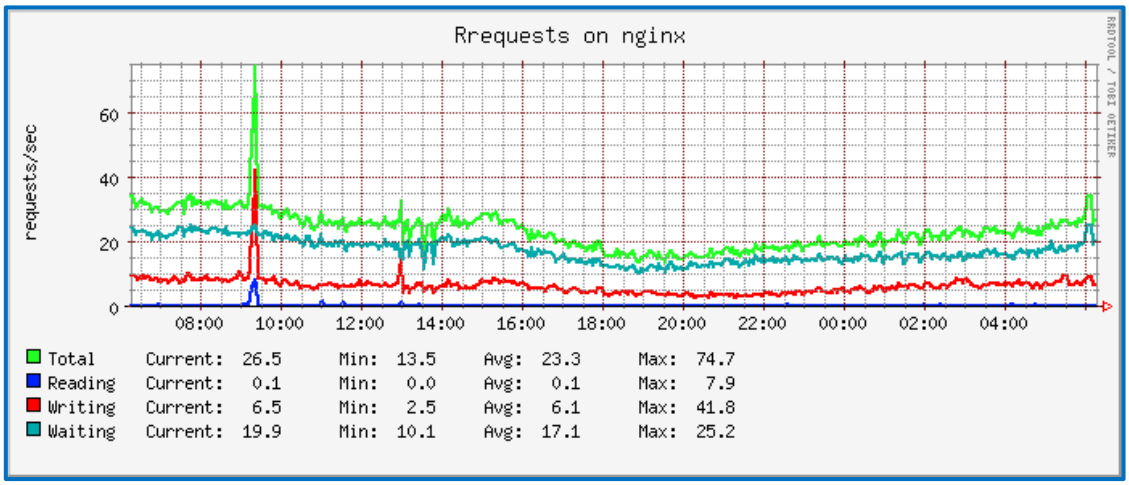
2. RRDTool

RRDTool (zkratka pro Round-Robin Database Tool) je v informatice opensource nástroj, který se zaměřuje na zpracování a ukládání časově závislých dat (například teplota, zatížení procesoru, síťový provoz a další). Tyto data jsou uložena v databázi typu round-robin, která má konstantní velikost v čase.

RRDTool je nová generace nástroje MRTG, obsahuje také nástroje pro získání dat v grafické podobě. Zásadní rozdíl oproti MRTG je, že se v principu již nejedná o úzce zaměřený program na sběr a vykreslení dat, jehož činnost je pouze modifikována parametry na příkazové řádce, ale o komplexní programovací prostředí pro archivaci a grafování dat, kterážto činnost se zcela řídí (i velmi dlouhým) programem zadaným v parametru příkazu. Lze zde oproti MRTG daleko snadněji dosáhnout stavu kdy je vygenerován prázdný graf aniž by přitom program vypsal nějakou chybovou hlášku, což může být velmi matoucí pro neprogramátory. Dále oproti MRTG chybí SNMP modul na sběr dat ze sítě [11].

2.1. Postup při vytvoření databáze a grafů

1. vytvoření prázdné databáze pomocí `RRDTool create`
2. pomocí skriptu a/nebo cronu opakovaně přidávat data do databáze pomocí `RRDTool update`
3. vytvořit, obvykle pomocí skriptu, jakkoliv modifikovaný graf pomocí `RRDTool graph`[11]¨



Obrázek 2.1: *Graf vytvořený pomocí nástroje RRDTool*

3. Instalace a konfigurace NetFlow na OS Linux

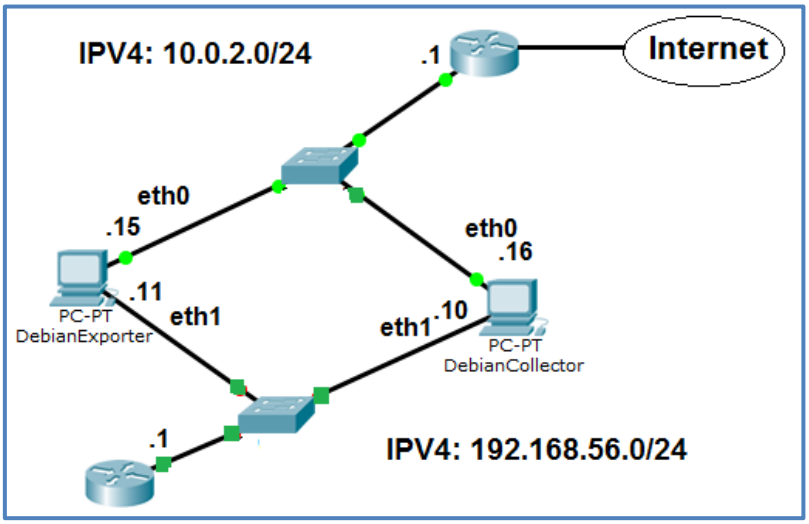
Cílem této části bylo otestovat protokol NetFlow na linuxových distribucích Debian a Ubuntu. Jednalo se o distribuci Ubuntu ve verzi 14.04 a distribuci Debian ve verzi 7.9 (wheezy). Všechny servery byly virtualizovány v prostředí Oracle VM Virtualbox. Protokol NetFlow byl odzkoušen na protokolech IPv4 a IPv6. Pro export a sběr datových toků byly použity open sourcové programy, použitelné na linuxové distribuce. Podrobnější postup instalace, konfigurace, výpisů a grafů bude následovat v dalších částech.

3.1. Konfigurace NetFlow na IPv4 - Linux Debian

Pro konfiguraci NetFlow nad protokolem IPv4 jsem zvolil již zmíněný Debian 7.9 (wheezy). Protokol NetFlow jako takový byl použit ve verzi 9.

3.1.1. Topologie

Topologie se skládala ze dvou virtualizovaných PC. Jeden byl exportér - měl 2 síťové rozhraní eth0 a eth1. Pomocí eth0 měl přístup na internet a pomocí eth1 byl spojen s druhým virtualizovaným PC - kolektorem. Funkcí exportéru bylo sbírat datové toky na rozhraní **eth0**, které pak byly poslány na kolektor. Kolektor měl pouze jedno síťové rozhraní eth1, pomocí kterého byl spojen s exportérem. Úkolem kolektoru byl sběr NetFlow dat, poslaných z exportéru. Nastavené IP adresy na rozhraních jsou uvedené na obrázku níže.



Obrázek 3.1: *Topologie sítě*

3.1.2. Instalace a konfigurace na exportéru (DebianExporter)

Jako NetFlow exportér byl použit program **Softflowd** ve verzi 0.9.9. Tento program jsem zvolil z důvodu podpory NetFlow verze 9 (podpora IPv4 a IPv6 protokolu). Instalaci provedeme příkazem:

```
apt-get update
apt-get install softflowd
```

Důležité je taky nastavit správný název serveru - v našem případě DebianExporter:

```
echo "DebianExporter" > /etc/hostname
```

Rovněž je nutné editovat soubor /etc/hosts:

```
127.0.0.1 localhost
192.168.56.11   DebianExporter
192.168.56.10   DebianCollector
```

Obrázek 3.2*: Výpis lokální DNS kolektoru a exportéru*

Nyní je nutno nastavit síťové rozhraní eth1 na IP adresu 192.168.56.11. Na rozhraní eth0 je již automaticky nastavena adresa 10.0.2.15, pomocí které máme přístup k internetu, tudíž tam adresu nastavovat nemusíme. Nastavení IP adresy na rozhraní eth1 provedeme příkazem:

```
ip a a 192.168.56.11/24 dev eth1
```

Po nakonfigurování síťového rozhraní můžeme Softlowd zapnout příkazem níže. Parametr -D spouští debugovací režim, parametr -v spouští NetFlow ve verzi v našem případě 9, parametr -i definuje, na kterém síťovém rozhraní bude exportér zachytávat provoz (v našem případě **eth0**, tedy síťové rozhraní, pomocí kterého máme přístup na internet), parametr -n definuje, na jakou IP adresu se budou výsledné datové toky posílat a na jaký port, pomocí parametru -T budem na kolektor posílat všechny informace o daném datovém toku a poslední parametr -m definuje, kolik datových toků lze maximálně sledovat před odesláním na kolektor. Tento parametr tse zde nachází z důvodu častého odesílání NetFlow dat na kolektor. Pokud by tam ten parametr nebyl, mohlo by se stávat, že na kolektor by byly odesílané informace o datových tocích, které začaly a skončily například před pěti minutama.

```
softflowd -D -v 9 -i eth0 -n 192.168.56.10:9995 -T full -m 10
```

Nyní je nutno nějaký provoz generovat. Pro to jsem použil na exportéru prohlížeč "Iceweasel Web Browser". Výhodou použití prohlížeče před klasickými paketovými generátory je možnost otestování v reálném provozu.

3.1.3. Instalace a konfigurace na kolektoru (DebianCollector)

Na NetFlow kolektoru byl použit program **Nfdump** ve verzi 1.6.6, který obsahuje nástroje pro zachycení datových toků - **Nfcapd** a pro analýzu datových toků - **Nfdump**. Výstupem bude HTML stránka s grafy - je nutno ještě nainstalovat webový server **Apache2** (verze 2.2.22) a nástroj pro tvorbu grafu **RRDTool** (verze 1.4.7). Instalaci provedeme pomocí příkazů:

```
apt-get update
apt-get install nfdump
apt-get install apache2
apt-get install rrdtool
```

Důležité je taky nastavit správný název serveru - v našem případě DebianCollector:

```
echo "DebianCollector" > /etc/hostname
```

Rovněž je nutné editovat soubor /etc/hosts:

```
127.0.0.1 localhost
192.168.56.11   DebianExporter
192.168.56.10   DebianCollector
```

Obrázek 3.3: Výpis lokální DNS kolektoru a exportéru

Nyní je nutno nastavit síťové rozhraní **eth1** na IP adresu 192.168.56.10. To provedeme příkazem:

```
IP a a 192.168.56.10/24 dev eth1
```

Po nakonfigurování síťového rozhraní můžeme sběr datových toků zapnout příkazem níže. Pomocí parametrů -w a -t se budou vytvářet každých 60 sekund nové soubory se zachyceným provozem. Parametr -D spouští program nfcapd na pozadí, pomocí parametru -p definujeme port, na kterém budeme ¨NetFlow data přijímat, parametr -n definuje, ze které stanice budeme datové toky přijímat a do které složky se budou ukládat.

```
nfcapd -w -t 60 -D -p 9995 \
-n DebianExporter,192.168.56.11,/home/student/flows
```

Jakmile je soubor se zachyceným provozem vytvořen, můžeme ho otevřít pomocí příkazu nfdump. Například:

```
nfdump -r nfcapd.201706041505
```

Dále je nutno vytvořit databázi v nástroji RRDTool. Maximální počet hodnot, které lze uložit do databáze bude 400, časový odstup mezi hodnotami v grafu bude 60 sekund (1 minuta), tzn. záznamy v databázi jsou uloženy maximálně pro 6 hodin a 40 minut. Maximální hodnota v grafu na svislé ose je 4000. Heartbeat, tj. maximální doba, po kterou nebudou do grafu vkládány nové hodnoty a graf zůstane zachován, je nastavena na 900 sekund (15 minut).

```
RRDTool create netflow.rrd --step 60 --start N \
DS:toky:GAUGE:900:0:4000 RRA:AVERAGE:0.5:1:400
```

3.1.4. Skript

Níže uvedený skript bude analyzovat poslední soubor analyzovaných datových toků, které byly přijaty z exportéru (DebianExporter). Tyto soubory se vytvářejí každou minutu. Je zde použit cyklus while, pomocí kterého se skript bude automaticky spouštět každou minutu.

V první části je třeba vybrat správný soubor - vybere se vždycky ten předposlední ve složce, protože do toho posledního souboru se data teprve ukládají. Následně se z daného souboru vypíšou pouze zdrojové a cílové porty a vyberou se pouze s číslem 80 (HTTP). Nyní se spočítá počet datových toků (jeden řádek je jeden datový tok), které byly vybrány a výsledná hodnota je zapsána do proměnné. Nyní je třeba aktualizovat hodnoty v grafu - hodnota, uložená v proměnné, obsahující počet datových toků na portu 80, je vložena do RRDTool databáze.

```
#!/bin/bash

path=/var/www/

while true; do
cd /home/student/flows
ls -1 | grep nfcapd > soubory.txt
tail -n 2 soubory.txt > soubory_dva.txt
head -1 soubory_dva.txt > soubor_vysl.txt

nfdump -r `cat soubor_vysl.txt` -o "fmt:%sp%dp" | grep -w 80 | wc -l > netflowx

start_hour=$((`date +%s`-3600))
start_six=$((`date +%s`-21600))

rrdtool update netflow.rrd  N:`cat netflowx`
```

Obrázek 3.4: *První část skriptu*

Po aktualizaci hodnot v RRDTool databázi lze přistoupit k samotnému vykreslování grafů. Uvedená část skriptu popisuje tvorbu šestihodinového grafu, pro hodinový graf je příkaz téměř totožný, liší se pouze v proměnné pro začátek grafu **(--start)** a v popisu grafu. Obrázky s grafy se budou ukládat do adresáře **/var/www/**, kde je již vytvořena stránka v HTML, ve které se budou obrázky zobrazovat. Popis dalších parametrů a atributů grafu:

- --title - titulek grafu
- --start, --end - časový začátek a konec grafu, zde je začátek o 6 hodin zpět
- --vertical-label - popis svislé osy
- -w, -h - parametry šířky a výšky grafu v pixelech
- DEF - definuje proměnné vyčtené z RRDTool databáze
- VDEF - vytváříme novou proměnnou, kterou pak využijeme u časového razítka
- LINE2 - vykreslí čáru tlustou 2 px v barvě, která je zadaná v hexadecimální soustavě
- GPRINT - vykreslí nám hodnoty dat pod graf (minimum, maximum, průměr), každá z hodnot bude mít přiřazený barevný čtvereček [16]

Na konci jsou soubory pro ukládání pomocných proměnných smazány a cyklus while je na 1 minutu pozastaven. Pak se spustí znovu.

```
rrdtool graph `echo $path`netflow6.png --title "Šestihodinový graf" --start $start_six \
--end N --vertical-label "Počet datových toků" -w 700 -h 300 \
DEF:toky=netflow.rrd:toky:AVERAGE \
VDEF:last=toky,LAST \
AREA:toky#66ff33 \
COMMENT:"------------------------------------------------------------\\n" \
LINE2:toky#58FAF4:"Minimální počet datových toků\:" \
GPRINT:toky:MIN:"%5.01f" \
COMMENT:"\\n" \
LINE2:toky#FF0000:"Maximální počet datových toků\:" \
GPRINT:toky:MAX:"%5.01f" \
COMMENT:"\\n" \
LINE2:toky#9595FF:"Průměrný počet datových toků\:" \
GPRINT:toky:AVERAGE:"%5.01f" \
LINE2:toky#CC0099 \
COMMENT:"\\n" \
COMMENT:"------------------------------------------------------------\\n" \
GPRINT:last:"Poslední aktualizace\: %c":strftime

rm soubory.txt
rm soubory_dva.txt
rm soubor_vysl.txt
rm netflowx

sleep 60
done
```

Obrázek 3.5*: Druhá část skriptu*

3.1.5. Nasbírané NetFlow data

Následující obrázek obsahuje defaultní výpis zachycených datových toků (je vypsáno pouze prvních 6 řádků ze souboru).. Je zde časové razítko začátku datového toku, zdrojové IP adresy, cílové IP adresy, porty, počet přenesených paketů, počet přenesených bytů, počet datových toků. Je možný také rozšířený výpis, kde můžou být vypsány další parametry datového toku.

```
root@DebianExporter:/home/student# nfdump -r nfcapd.201704061702 -c 6
Date flow start          Duration Proto      Src IP Addr:Port          Dst IP Addr:Port   Packets    Bytes Flows
2017-04-06 17:02:54.957     2.014 TCP          10.0.2.15:36657 ->    216.58.201.98:443        11     3563     1
2017-04-06 17:02:54.957     2.014 TCP      216.58.201.98:443   ->        10.0.2.15:36657       13     6540     1
2017-04-06 17:02:57.482     0.029 TCP          10.0.2.15:60035 ->   85.232.230.227:443          3      173     1
2017-04-06 17:02:57.482     0.029 TCP     85.232.230.227:443   ->        10.0.2.15:60035        3      138     1
2017-04-06 17:02:57.467     5.755 TCP          10.0.2.15:41651 ->      23.37.43.27:80           4      180     1
2017-04-06 17:02:57.467     5.755 TCP        23.37.43.27:80    ->        10.0.2.15:41651        3      138     1
Summary: total flows: 6, total bytes: 10732, total packets: 37, avg bps: 10387, avg pps: 4, avg bpp: 290
Time window: 2017-04-06 17:02:54 - 2017-04-06 17:03:50
Total flows processed: 66, Blocks skipped: 0, Bytes read: 3996
Sys: 0.000s flows/second: 0.0        Wall: 0.000s flows/second: 79903.1
```

Obrázek 3.6: *Výpis datových toků na kolektoru (DebianCollector)*

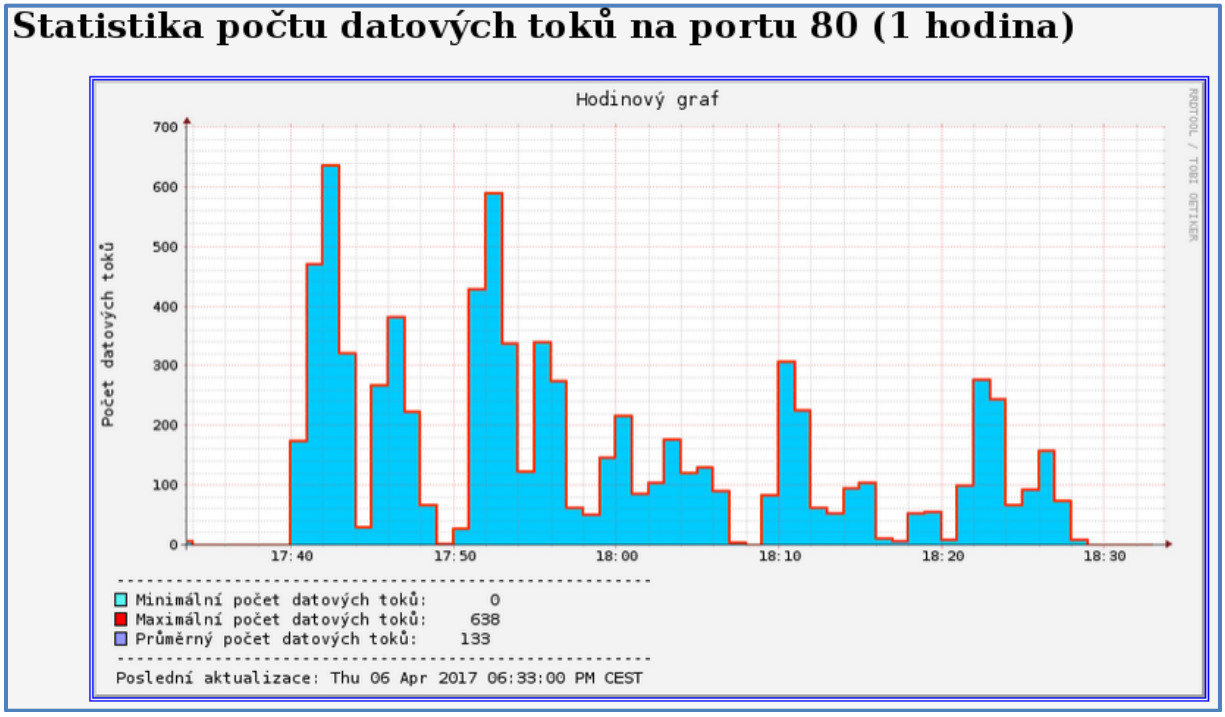
Druhý výpis obsahuje vyfiltrované zdrojové a cílové porty analyzovaných datových toků (je vypsáno pouze prvních 6 řádků ze souboru). Pomocí nástroje **grep** jsou z tohoto výpisu vybrány pouze řádky, které obsahují port 80 a ty jsou následně spočítány.

```
root@DebianExporter:/home/student# nfdump -r nfcapd.201704061702 -c 6 -o "fmt:%sp%dp"
Src PtDst Pt
 36657   443
   443 36657
 60035   443
   443 60035
 41651    80
    80 41651
```

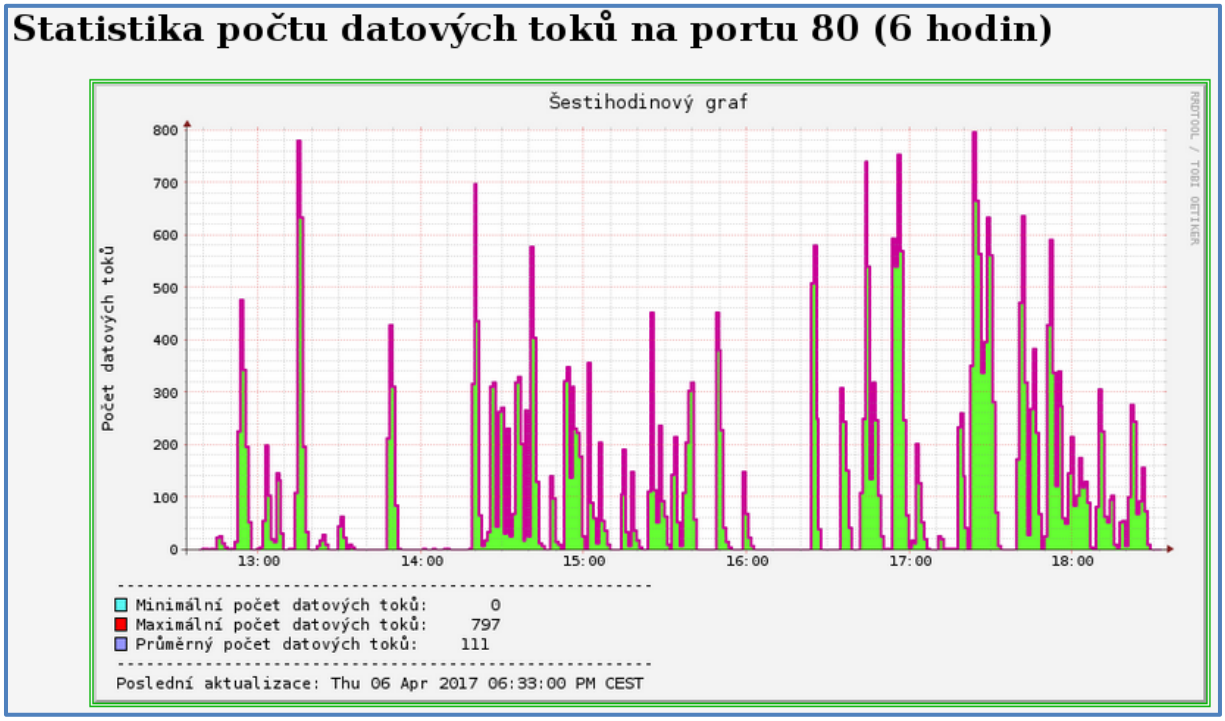
Obrázek 3.7: *Výpis portů z datových toků na kolektoru (DebianCollector)*

3.1.6. Grafy

Výstupem skriptu jsou 2 grafy. Jeden hodinový a druhý šestihodinový. Grafy ukazují počet datových toků na portu 80. Pod grafem je rovněž uveden maximální, minimální, průměrný počet datových toků a datum a čas poslední aktualizace. Grafy byly zobrazovány v HTML stránce.



Obrázek 3.8*: Hodinový graf zobrazující počet datových toků na portu 80*



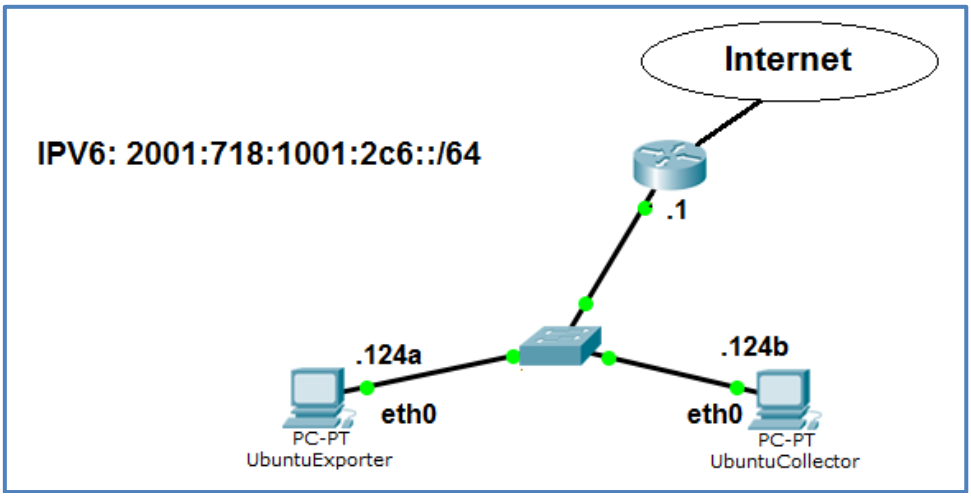
Obrázek 3.9: *Šestihodinový graf zobrazující počet datových toků na portu 80*

3.2. Konfigurace NetFlow na IPv6 - Linux Ubuntu

Pro konfiguraci NetFlow nad IPv6 protokolem jsem zvolil již zmíněný Ubuntu 14.04. Protokol NetFlow byl použit ve verzi 9 - tato verze byla zde zvlášť potřebná, protože podporuje IPv6 protokol.

3.2.1. Topologie

Topologie se skládala ze dvou virtualizovaných PC. Jeden byl exportér (UbuntuExporter) s rozhraním eth0, na kterém byla nastavena IPv6 adresa. Funkcí exportéru bylo sbírat datové toky na rozhraní eth0, které pak byly poslány na kolektor. Kolektor (UbuntuCollector) měl také jedno síťové rozhraní **eth0** s nastavenou IPv6 adresou. Úkolem kolektoru byl sběr NetFlow dat, poslaných z exportéru. Nastavené IP adresy na rozhraních a názvy jednotlivých počítačů jsou uvedené na obrázku 3.10.



Obrázek 3.10: *Topologie sítě*

3.2.2. Instalace a konfigurace na exportéru (UbuntuExporter)

Pro export datových toků byl použit jíž zmíněný program **Softflowd** ve verzi 0.9.9. Instalaci provedeme příkazem:

```
apt-get update
apt-get install softflowd
```

Důležité je taky nastavit správný název serveru - v našem případě UbuntuExporter:

```
echo "UbuntuExporter" > /etc/hostname
```

Rovněž je nutné editovat soubor /etc/hosts:

```
127.0.0.1 localhost
2001:718:1001:2c6::124b  UbuntuCollector
2001:718:1001:2c6::124a  UbuntuExporter
```

Obrázek 3.11: Výpis lokální DNS kolektoru a exportéru

Po nakonfigurování souboru hosts můžeme spustit program Softlowd. Pomocí parametru -6 definujeme, že cílový protokol pro analýzu provozu bude IPv6, parametr -D spouští Softflowd v debugovacím režimu, parametr -v spouští NetFlow ve verzi v našem případě 9, parametr -i definuje, na kterém rozhraní bude exportér zachytávat provoz (v našem případě **eth0**), parametr -n definuje, na jakou IP adresu se budou výsledné datové toky posílat a na jaký port, pomocí parametru -T budem na kolektor posílat všechny informace o daném datovém toku a poslední parametr -m definuje, kolik datových toků lze maximálně sledovat před odesláním na kolektor.

```
softflowd -6 -D -v 9 -i eth0 -n 2001:718:1001:2c6::124b:9995 \
-T full -m 10
```

Nyní je nutno nějaký provoz generovat. Protože jsme na stanici, kde není nainstalovené GUI, použijeme prohlížeč v příkazovém řádku. Nainstaloval jsem si prohlížeč **w3m**, přes který jsem se pohyboval na různých stránkách a tím generoval provoz na rozhraní **eth0**.

3.2.3. Instalace a konfigurace na kolektoru (UbuntuCollector)

Na NetFlow kolektoru byl použit program Nfdump ve verzi 1.6.8, který obsahuje jak program pro zachycení datových toků - Nfcapd, tak program pro analýzu - Nfdump. Výstupem bude HTML stránka s grafy - je nutno ještě nainstalovat webový server Apache2 (verze 2.4.7) a nástroj pro tvorbu grafu RRDTool (verze 1.4.7). Instalaci provedeme pomocí příkazů:

```
apt-get update
apt-get install nfdump
apt-get install apache2
apt-get install rrdtool
```

Důležité je taky nastavit správný název serveru - v našem případě UbuntuCollector:

```
echo "UbuntuCollector" > /etc/hostname
```

Rovněž je nutné editovat soubor /etc/hosts:

```
127.0.0.1 localhost
2001:718:1001:2c6::124b  UbuntuCollector
2001:718:1001:2c6::124a  UbuntuExporter
```

Obrázek 3.12: Výpis lokální DNS kolektoru a exportéru

Nyní můžeme sběr datových toků zapnout příkazem níže Pomocí parametru -6 definujeme, že cílový protokol pro analýzu provozu bude IPv6. Pomocí parametrů -w a -t se budou vytvářet každých 60 sekund nové soubory se zachyceným provozem. Parametr -D spouští program nfcapd na pozadí, pomocí parametru -p definujeme port, na kterém budeme NetFlow data přijímat, parametr -n definuje, ze které stanice budeme datové toky přijímat a do které složky se budou ukládat. .

```
nfcapd -6 -w -t 60 -D -p 9995
-n UbuntuExporter, 2001:718:1001:2c6::124a,/home/student/flows
```

Jakmile je soubor se zachyceným provozem vytvořen, můžeme ho analyzovat. To provedeme pomocí příkazu nfdump:

```
nfdump -r nfcapd.201704061715
```

Dále je nutno vytvořit databázi v nástroji RRDTool. Maximální počet hodnot, které lze uložit do databáze bude 400, časový odstup mezi hodnotami v grafu bude 60 sekund (1 minuta), tzn. záznamy v databázi jsou uloženy maximálně pro 6 hodin a 40 minut. Maximální hodnota v grafu na svislé ose je 4000. Heartbeat, tj. maximální doba, po kterou nebudou do grafu vkládány nové hodnoty a graf zůstane zachován, je nastavena na 900 sekund (15 minut).

```
RRDTool create netflow.rrd --step 60 --start N \
DS:toky:GAUGE:900:0:4000 RRA:AVERAGE:0.5:1:400
```

3.2.4. Skript

Skript pro zpracovávání provozu a vykreslování grafů je stejný jako ten, který byl použit na počítačích s operačním systémem Linux Debian. Jsou zde pouze odlišnosti v cestě k webovému adresáři a v jiném portu, podle kterého se vybírají datové toky. Oproti předchozí kapitole, zde se vybírají všechny datové toky, které obsahují zdrojový nebo cílový port 443 (HTTPS).

```
#!/bin/bash

path=/var/www/html/

while true; do
cd /home/student/flows
ls -1 | grep nfcapd > soubory.txt
tail -n 2 soubory.txt > soubory_dva.txt
head -1 soubory_dva.txt > soubor_vysl.txt

nfdump -r `cat soubor_vysl.txt` -o "fmt:%sp%dp" | grep -w 443 | wc -l > netflowx
```

Obrázek 3.13: *První část skriptu*

3.2.5. Výpis z kolektoru (UbuntuCollector)

Níže uvedený výpis obsahuje vyfiltrované porty analyzovaných datových toků (je vypsáno pouze prvních 5 řádků ze souboru). Pomocí nástroje **grep** jsou z tohoto výpisu vybrány pouze řádky, které obsahují port 443 a ty jsou následně spočítány.

```
root@UbuntuCollector:/home/student/flows# nfdump -r nfcapd.201704061700 -c 5 -o "fmt:%sp%dp"
Src PtDst Pt
    53 41565
 42676   443
   443 42676
    53 42911
 42911    53
Summary: total flows: 5, total bytes: 21047, total packets: 28, avg bps: 21855, avg pps: 3, avg bpp: 751
```

Obrázek 3.14: *Výpis portů z datových toků na kolektoru (UbuntuCollector)*

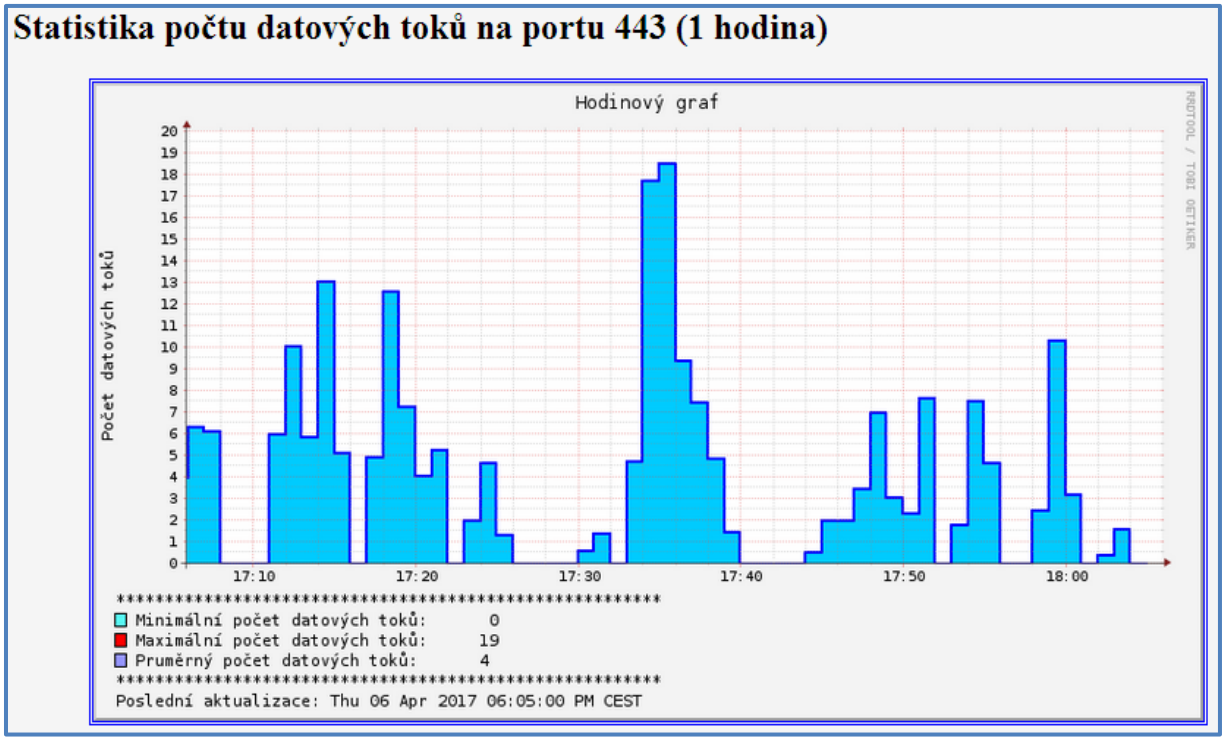
Druhý obrázek ukazuje defaultní výpis zachycených datových toků (je vypsáno pouze prvních 10 řádků ze souboru).. Je zde časové razítko začátku datového toku, zdrojové IPv6 adresy, cílové IPv6 adresy, porty, počet přenesených paketů, počet přenesených bytů, počet datových toků. Samozřejmě je možné si vypsat více parametrů datových toků (například časové razítko konce datového toku). Lze si také všimnout, že část IPv6 adresy je useknutá. Pro zobrazení celé IPv6 adresy by bylo přidat do příkazu **nfdump** parametr **-o line6.**

```
root@UbuntuCollector:/home/student/flows# nfdump -r nfcapd.201704061700 -c 10
Date flow start          Duration Proto      Src IP Addr:Port          Dst IP Addr:Port   Packets    Bytes Flows
2017-04-06 17:00:39.512     0.000 UDP   2001:71..:149::9.53    -> 2001:71..6::124a.41565        2      462     1
2017-04-06 17:00:39.509     0.074 TCP   2001:71..6::124a.42676 -> 2a02:59..::78:63.443         11     1632     1
2017-04-06 17:00:39.509     0.074 TCP   2a02:59..::78:63.443   -> 2001:71..6::124a.42676       11    18331     1
2017-04-06 17:00:47.190     0.023 UDP   2001:71..:149::9.53    -> 2001:71..6::124a.42911        2      464     1
2017-04-06 17:00:47.190     0.023 UDP   2001:71..6::124a.42911 -> 2001:71..:149::9.53           2      158     1
2017-04-06 17:00:47.259     0.001 UDP   2001:71..:149::9.53    -> 2001:71..6::124a.56029        2      464     1
2017-04-06 17:00:47.259     0.001 UDP   2001:71..6::124a.56029 -> 2001:71..:149::9.53           2      158     1
2017-04-06 17:00:47.542     0.049 TCP   2001:71..6::124a.54126 -> 2a02:59..:79:220.443         10     1579     1
2017-04-06 17:00:47.542     0.049 TCP   2a02:59..:79:220.443   -> 2001:71..6::124a.54126        9     9212     1
2017-04-06 17:01:18.008     0.000 ICMP6 fe80::2..55:a400.0     ->          ff02::1.0.0          1      104     1
Summary: total flows: 10, total bytes: 32564, total packets: 52, avg bps: 6766, avg pps: 1, avg bpp: 626
Time window: 2017-04-06 17:00:39 - 2017-04-06 17:01:23
Total flows processed: 16, Blocks skipped: 0, Bytes read: 1876
Sys: 0.007s flows/second: 2029.9     Wall: 0.001s flows/second: 9019.2
```

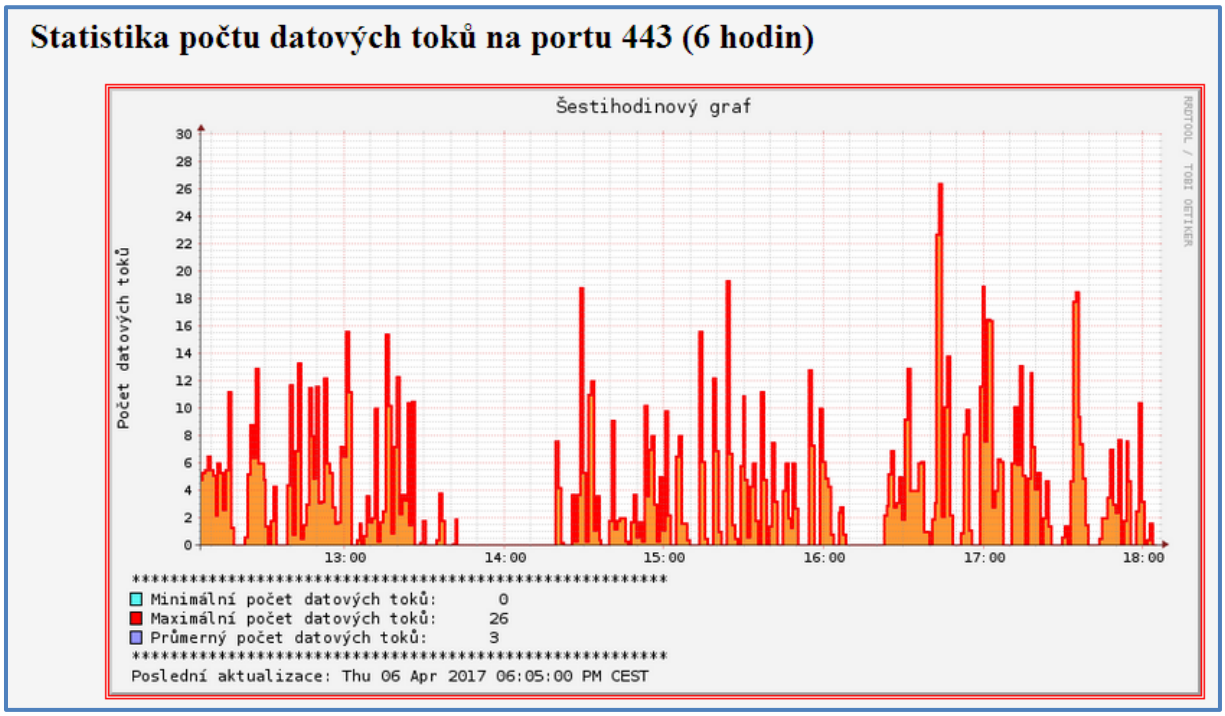
Obrázek 3.15: Výpis datových toků na kolektoru (UbuntuCollector)

3.2.6. Grafy

Výstupem byly dva grafy - jeden hodinový a druhý šestihodinový, které byly zobrazovány v HTML stránce. Oba grafy obsahují nadpis, popis svislé osy, minimální, maximální, průměrný počet datových toků a časové razítko poslední aktualizace grafu



Obrázek 3.16: *Hodinový graf zobrazující počet datových toků na portu 443*



Obrázek 3.17: *Šestihodinový graf zobrazující počet datových toků na portu 443*

4. Konfigurace NetFlow na Cisco směrovačích

V této kapitole jsem otestoval NetFlow protokol na Cisco směrovačích, konkrétně se jednalo o směrovače Cisco 2800 s různými verzemi IOSu.

První část této kapitoly se věnuje konfiguraci na Cisco směrovačích s verzí IOSu 12.3.(14)YT. Tato verze umožňuje konfiguraci NetFlow pro sběr IPv4 a IPv6 datových toků. Kromě samotné konfigurace na Cisco směrovačích zde bude také popsána topologie síťě, konfigurace NetFlow kolektoru, výpisy tabulek na Cisco směrovačích, konfigurace nástroje pro tvorbu grafů RRDTool a skripty, které generují testovací provoz. Pro generování provozu jsem zvolil nástroj Mgen z důvodu jeho použitelnosti na IPv6 protokolu. Jako další jsem zvolil Mz (Mausezahn) a Hping3. Všechny tyto nástroje umí generovat TCP, UDP a ICMP provoz [12][13].

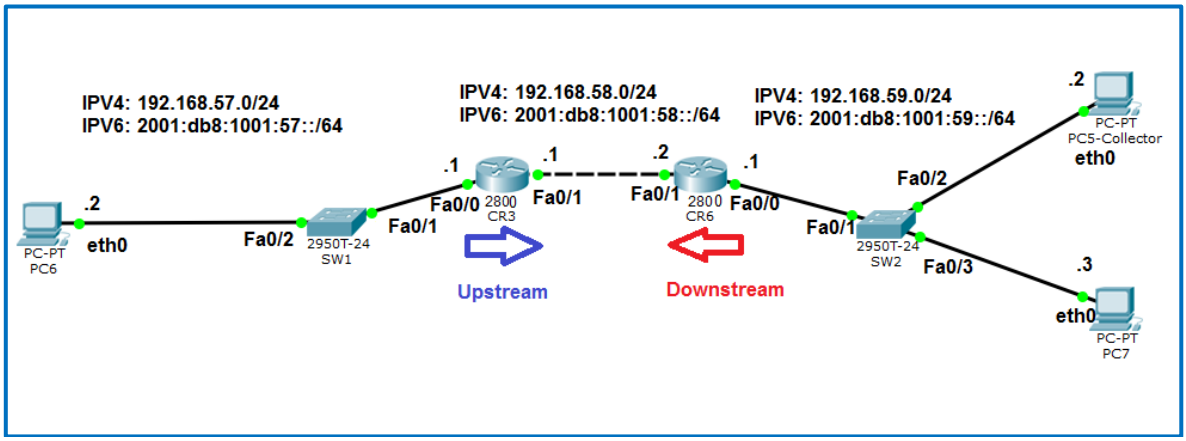
Druhá část této kapitoly se věnuje konfiguraci NetFlow na Cisco směrovačů s verzí IOSu 12.4(22)T. Tato verze je odlišná v konfiguraci NetFlow pro sběr IPv6 datových toků. Je zde zavedeno tzv. Flexibilní NetFlow, které Cisco nasadilo na směrovače od verze IOSu 12.4(20)T. Pro IPv4 protokol je konfigurace stejná, jako v předchozí kapitole. Dále zde bude také popsána topologie síťě, konfigurace NetFlow kolektoru, výpisy tabulek na Cisco směrovačích, konfigurace nástroje pro tvorbu grafů RRDTool a skripty, které generují testovací provoz [12][14][15].

Ve třetí části je popsána konfigurace Flexibilního NetFlow pro sběr IPv4 a IPv6 provozu. Oproti předchozím kapitolám, se zde neexportují na kolektor všechny parametry provozu, ale jen některé, které jsou předem definované. Díky tomu jsou sníženy výpočetní nároky na systémové zdroje směrovače [14] [15].

4.1. Konfigurace NetFlow na Cisco 2800 (IOS 12.3(14)yz1)

4.1.1. Topologie

Topologie se skládala ze tří počítačů s operačním systémem Ubuntu 14.04. Všechny počítače měly jedno síťové rozhraní, pomocí kterého byly připojeny do Cisco přepínačů. Dále se topologie skládá ze dvou Cisco směrovačů 2801, na kterých je nakonfigurován export datových toků vždycky pro jeden směr provozu (upstream, downstream, tj. každý směrovač měl na starosti jeden směr) a pro oba protokoly současně (IPv4 a IPv6). Aby bylo možné sbírat informace o provozu v obou směrech zvlášť, bylo nutno použít v topologii 2 směrovače -při použití klasického NetFlow protokolu by sice bylo možné sbírat datové toky z obou směrů, avšak nebylo by možné je pak oddělit na kolektoru. Při použití Flexibilního NetFlow stačí použít jeden směrovač, viz. kapitola 4.3. Jeden z počítačů měl funkci kolektoru (PC5_Collector) - sbíral NetFlow data od příslušných exportérů (Cisco směrovače). Topologie sítě s adresováním jednotlivých stanic je uvedena na obrázku níže [12][13].



Obrázek 4.1: *Topologie sítě*

4.1.2. Konfigurace na směrovači CR3

Nejdříve je nutno si pojmenovat směrovač (v našem případě CR3) a nakonfigurovat přeposílání IPv6 unicastových paketů. Aby bylo možno aktivovat sběr IPv6 provozu na rozhraních, je nutno také aktivovat CEF (Cisco Express Forwarding) pro IPv6.

```
Router(config)#hostname CR3
CR3(config)#IPv6 unicast-routing
CR3(config-if)#IPv6 cef
CR6#clock set 12:00:00 21 Mar 2017
```

Dále je nutno nakonfigurovat IPv4 a IPv6 adresy na jednotlivé rozhraní a aktivovat tato rozhraní. Uvedu zde konfiguraci adres pro jedno rozhraní směrovače, pro ostatní rozhraní je konfigurace odlišná pouze v adresách.

```
CR3(config)#interface FastEthernet 0/0
CR3(config-if)#ip address 192.168.57.1 255.255.255.0
CR3(config-if)#ip address 2001:db8:1001:57::1/64
CR3(config-if)#no shutdown
```

Po nakonfigurování adres aktivujeme směrovací algoritmus OSPF pro IPv4 a OSPVv3 pro IPv6 na jednotlivých rozhraních. Pro směrovač CR3 jsem zvolil router-id 1.1.1.1.

```
CR3(config)#router ospf 1
CR3(config-router)#network 192.168.57.0 0.0.0.255 area 0
CR3(config-router)#network 192.168.58.0 0.0.0.255 area 0

CR3(config)#ipv6 router ospf 1
CR3(config-router)#router-id 1.1.1.1

CR3(config)#interface FastEthernet 0/0
CR3(config-if)#ipv6 ospf 1 area 0

CR3(config)#interface FastEthernet 0/1
CR3(config-if)#ipv6 ospf 1 area 0
```

Nyní zbývá aktivovat sběr datových toků pro IPv4 a IPv6 protokol a jejich odesílání na kolektor, kde budou dále zpracovávány. IPv4 datové toky budou odesílány na adresu 192.168.59.2 (IP adresa kolektoru) na portu 5534. IPv6 datové toky budou rovněž odesílány na adresu 192.168.59.2 (IP adresa kolektoru). Číslo portu bude 5536. Při konfiguraci si lze vybral jakýkoliv port, na kterém neběží žádná jiná služba. Dále je nutno nakonfigurovat zdroj (IP adresu, rozhraní), ze kterého se budou datové toky odesílat kolektoru, zvolíme to samé rozhraní, na kterém je aktivován sběr toků. Pak nastavíme verzi NetFlow - v našem případě verze 9. A nakonec je třeba aktivovat na rozhraní sběr datových toků. Rovněž je důležité nastavit směr - je možný směr provozu do rozhraní (ingress) nebo ven z rozhraní (egress) nebo oba současně. Zvolil jsem směr do rozhraní (ingress), protože je žádoucí mít přehled i o provozu, který ještě nebyl směrovačem vyfiltrován. Tudíž tento směrovač bude sbírat datové toky pro upstream provoz.

```
CR3(config)#ip flow-export source FastEthernet 0/0
CR3(config)#ip flow-export version 9
CR3(config)#ip flow-export destination 192.168.59.2 5534

CR3(config)#ipv6 flow-export source FastEthernet 0/0
CR3(config)#ipv6 flow-export destination 192.168.59.2 5536

CR3(config)#interface FastEthernet 0/0
CR3(config-if)#ip flow ingress
CR3(config-if)#ipv6 enable
CR3(config-if)#ipv6 flow ingress
```

4.1.3. Konfigurace na směrovači CR6

Stejně jako u konfigurace CR3, i zde bude konfigurace podobná. Opět je nutno si pojmenovat směrovač (v našem případě CR6) a nakonfigurovat přeposílání IPv6 unicastových paketů. Aby bylo možno aktivovat sběr IPv6 provozu na rozhraních, je nutno také aktivovat CEF (Cisco Express Forwarding) pro IPv6. Rovněž je nutné nastavit správný čas na směrovači.

```
Router(config)#hostname CR6
CR6(config)#ipv6 unicast-routing
CR6(config-if)#ipv6 cef
CR6#clock set 12:00:00 21 Mar 2017
```

Dále je nutno nakonfigurovat IPv4 a IPv6 adresy na jednotlivé rozhraní a aktivovat tato rozhraní. Uvedu zde konfiguraci adres pro jedno rozhraní směrovače, pro ostatní rozhraní je konfigurace odlišná pouze v adresách.

```
CR6(config)#interface FastEthernet 0/0
CR6(config-if)#ip address 192.168.59.1 255.255.255.0
CR6(config-if)#ip address 2001:db8:1001:59::1/64
CR6(config-if)#no shutdown
```

Po nakonfigurování adres aktivujeme směrovací algoritmus OSPF pro IPv4 a OSPVv3 pro IPv6. Pro směrovač CR6 jsem zvolil router-id 2.2.2.2.

```
CR6(config)#router ospf 1
CR6(config-router)#network 192.168.58.0 0.0.0.255 area 0
CR6(config-router)#network 192.168.59.0 0.0.0.255 area 0

CR6(config)#ipv6 router ospf 1
CR6(config-router)#router-id 2.2.2.2

CR6(config)#interface FastEthernet 0/0
CR6(config-if)#ipv6 ospf 1 area 0

CR6(config)#interface FastEthernet 0/1
CR6(config-if)#ipv6 ospf 1 area 0
```

Nyní zbývá aktivovat sběr datových toků pro IPv4 a IPv6 protokol a jejich odesílání na kolektor, kde budou dále zpracovávány. IPv4 datové toky budou odesílány na adresu 192.168.59.2 (IP adresa kolektoru) na portu 5564. IPv6 datové toky budou rovněž odesílány na adresu 192.168.59.2 (IP adresa kolektoru). Číslo portu bude 5566. Jako zdroj (IP adresu, rozhraní), ze kterého se budou datové toky odesílat kolektoru, zvolíme to samé rozhraní, na kterém je aktivován sběr toků. Pak nastavíme verzi NetFlow - v našem případě verze 9. A nakonec je třeba aktivovat na rozhraní sběr datových toků. Zvolil jsem směr provozu, který vstupuje do rozhraní (ingress). Tudíž tento směrovač bude sbírat datové toky pro downstream provoz.

```
CR6(config)#ip flow-export source FastEthernet 0/0
CR6(config)#ip flow-export version 9
CR6(config)#IP flow-export destination 192.168.59.2 5564

CR6(config)#ipv6 flow-export source FastEthernet 0/0
CR6(config)#ipv6 flow-export destination 192.168.59.2 5566

CR6(config)#interface FastEthernet 0/0
CR6(config-if)#ip flow ingress
CR6(config-if)#ipv6 enable
CR6(config-if)#ipv6 flow ingress
```

4.1.4. Konfigurace síťových rozhraní na počítačích

Po nakonfigurování směrovačů je třeba nastavit adresy na rozhraní jednotlivých počítačů a otestovat dostupnost pomocí příkazu ping, popř. ping6. Konfiguraci opět uvedu pro jeden počítač (PC5-Collector), pro ostatní počítače bude odlišna pouze v adresách. Skládá se z nastavení IPv4 a IPv6 adresy na rozhraní a přidání defaultní brány.

```
ip a a 192.168.59.2/24 dev eth0
ip a a 2001:db8:1001:59::2/64 dev eth0
ip route add default via 192.168.59.1

ping host
ping6 host
```

4.1.5. Instalace a konfigurace na kolektoru (PC5-Collector)

Instalace

Pro sběr NetFlow dat byl zvolen program Nfdump (verze 1.6.8), který obsahuje jak program pro zachycení datových toků - Nfcapd, tak program pro analýzu - Nfdump. Jeho výhodou je, že umí analyzovat IPv4 a IPv6 provoz. Dále nainstalujeme program RRDTool (1.4.7) pro ukládání hodnot a vytváření grafů. Grafy se budou zobrazovat na webu, tudíž bude nutno nainstalovat webový server Apache2 (2.4.7). A nakonec se nainstalují programy pro generování provozu (Mgen, Mausezahn).

```
apt-get update
apt-get install nfdump
apt-get install RRDTool
apt-get install apache2
apt-get install mgen
apt-get install mz
```

Po instalaci příslušných programů je třeba vytvořit adresáře, do kterých se budou ukládat soubory se zachyceným provozem. Jedná se o čtyři adresář, v každé budou uloženy buď soubory s upstream provozem pro IPv4 nebo IPv6 protokol, nebo soubory s downstream provozem pro IPv4 nebo IPv6 protokol. Dále vytvoříme adresář, ve které budou uloženy webové stránky a obrázky grafů. Zobrazování grafů na webu mají na starost tři HTML stránky. Všem adresářům je vhodné přiradit příslušná práva pro zápis a čtení.

```
cd /home/student
mkdir flowsUpstreamIPV4
mkdir flowsUpstreamIPV6
mkdir flowsDownstreamIPV4
mkdir flowsDownstreamIPV6
chmod 775 flowsUpstreamIPV*
chmod 775 flowsDownstreamIPV*

cd /var/www/html/
mkdir NetFlow
chmod 775 NetFlow

cd /home/student/
```

Vytvoření databáze v RRDTool

Následně vytvoříme RRDTool databázi pro ukládání hodnot. Databáze bude vytvořena s krokem 60 sekund. Počátek je nastaven na N "ted"' (počet sekund od 1.1. 1970, kdy se začnou do databáze zapisovat první hodnoty). Obsahuje 36 proměnný typu GAUGE - typ proměnné, u které to, co je zadáno, to se vypíše. Maximální délka názvu proměnné může být 19 znaků. Heartbeat (čas, po který nebudou chodit hodnoty a do grafu se nic vypisovat nebude) je nastaven na 900 sekund (15 minut). Minimální hodnota v grafu je 0 a maximální hodnota 500000. V posledním řádku definujeme, kolik kroků si bude databáze pamatovat pro každou proměnnou. V našem případě si databáze bude pamatovat 1500 kroků (jedná se o 1500 minut, to odpovídá 25 hodinám) [16].

```
RRDTool create flows.rrd --step 60 --start N
DS:pocetBytuUpIPV4TCP:GAUGE:900:0:500000
DS:pocetBytuUpIPV4UDP:GAUGE:900:0:500000
DS:pocetBytuUpIPV4ICMP:GAUGE:900:0:500000
DS:pocetBytuDoIPV4TCP:GAUGE:900:0:500000
DS:pocetBytuDoIPV4UDP:GAUGE:900:0:500000
DS:pocetBytuDoIPV4ICMP:GAUGE:900:0:500000
DS:pocetPakeUpIPV4TCP:GAUGE:900:0:500000
DS:pocetPakeUpIPV4UDP:GAUGE:900:0:500000
DS:pocetPakeUpIPV4ICMP:GAUGE:900:0:500000
DS:pocetPakeDoIPV4TCP:GAUGE:900:0:500000
DS:pocetPakeDoIPV4UDP:GAUGE:900:0:500000
DS:pocetPakeDoIPV4ICMP:GAUGE:900:0:500000
DS:pocetTokuUpIPV4TCP:GAUGE:900:0:500000
DS:pocetTokuUpIPV4UDP:GAUGE:900:0:500000
DS:pocetTokuUpIPV4ICMP:GAUGE:900:0:500000
DS:pocetTokuDoIPV4TCP:GAUGE:900:0:500000
DS:pocetTokuDoIPV4UDP:GAUGE:900:0:500000
DS:pocetTokuDoIPV4ICMP:GAUGE:900:0:500000
DS:pocetBytuUpIPV6TCP:GAUGE:900:0:500000
DS:pocetBytuUpIPV6UDP:GAUGE:900:0:500000
DS:pocetBytuUpIPV6ICMP:GAUGE:900:0:500000
DS:pocetBytuDoIPV6TCP:GAUGE:900:0:500000
DS:pocetBytuDoIPV6UDP:GAUGE:900:0:500000
DS:pocetBytuDoIPV6ICMP:GAUGE:900:0:500000
DS:pocetPakeUpIPV6TCP:GAUGE:900:0:500000
DS:pocetPakeUpIPV6UDP:GAUGE:900:0:500000
DS:pocetPakeUpIPV6ICMP:GAUGE:900:0:500000
DS:pocetPakeDoIPV6TCP:GAUGE:900:0:500000
DS:pocetPakeDoIPV6UDP:GAUGE:900:0:500000
DS:pocetPakeDoIPV6ICMP:GAUGE:900:0:500000
DS:pocetTokuUpIPV6TCP:GAUGE:900:0:500000
DS:pocetTokuUpIPV6UDP:GAUGE:900:0:500000
DS:pocetTokuUpIPV6ICMP:GAUGE:900:0:500000
DS:pocetTokuDoIPV6TCP:GAUGE:900:0:500000
DS:pocetTokuDoIPV6UDP:GAUGE:900:0:500000
DS:pocetTokuDoIPV6ICMP:GAUGE:900:0:500000
RRA:AVERAGE:0.5:1:1500
```

Konfigurace skriptu

Vzhledem k tomu, že skript je velmi rozsáhlý (přes 1200 řádků), popíšu zde ty nejdůležitější části - ostatní části jsou víceméně podobné a liší se akorát v parametrech. Celý skript je uveden v příloze.

Skript jako takový je napsán v jazyce Bash. Jeho hlavním úkolem je vybrat ze souborů, zachycených NetFlow kolektorem, TCP, UDP a ICMP provoz, pro každý z těchto provozů spočítat celkový počet přenesených kB, paketů a datových toků. Tyto hodnoty jsou zapsány do RRDTool databáze, ze které jsou následně vybírány a jsou z nich vytvářeny grafy. Tento skript je cyklicky spouštěn každých 50 sekund (když se započítá i zpoždění pro výpočet, tak se spouštění opakuje každých) - grafy jsou tedy aktualizovány každou minutu. I když soubory se zachyceným provozem se

vytvářejí každých 5 minut, je žádoucí nastavit aktualizaci grafu na dobu menší než zmíněných 5 minut - po náhlé změně hodnoty v grafu (v řádech stovek, tisíců) RRDTool neumí ihned vykreslit přesnou hodnotu, ale jen přibližnou s určitou odchylkou, po následné aktualizaci grafů je pak většinou už vypsána přesná hodnota.

Na začátku skriptu je třeba si uložit cestu adresáře do proměnné. V tomto adresáři se budou vytvářet fotografie grafů. Pak je spuštěn nekonečný cyklus while (skript je pak třeba vypnout manuálně) a dochází k získání hodnot ze zachycených souborů. Na obrázku níže je část skriptu, jejíž úkolem je ze zachycených souborů získat celkový počet bytů pro TCP provoz. Nejdříve se přesuneme do příslušného adresáře, v našem případě se jedná o adresář pro upstream provoz a protokol IPv4. Zde jsou každých 5 minut vytvářeny soubory se zachyceným provozem, který byl poslán z příslušného Cisco směrovače. Je nutno vybrat vždycky ten předposlední soubor (soubory jsou označené časovým razítkem a seřazené podle data vytvoření, např. **nfcapd.201703201530**), protože poslední je vždycky soubor, který se teprve vytváří (název nfcapd.current). Jakmile je název správného souboru vybrán, je uložen do výsledného textového souboru "**soubor_vysl.txt**", se kterým se bude následně pracovat. Pak je nutno nastavit na hodnotu 0 proměnnou, do které se budou počítat byty (respektive kB). Pomocí příkazu nfdump s příslušnými parametry je ze souboru se zachyceným provozem vypsán pouze sloupec s typem provozu (TCP, UDP, ICMP) a sloupec s počtem bytů. Z toho je pak vybrán pomocí nástroje grep pouze TCP provoz a pomocí nástroje awk je zapsán do souboru **vypisTCP** výsledný sloupec s počtem bytů. Ten je následně zpracován v cyklu for. Po každém průchodu je hodnota vydělena číslem 1024 (převod na kB), výsledné číslo je zároveň převedeno na datový typ float na 3 desetinná místa a následně je přičteno. Po provedení cyklu for je proměnná, která představuje součet hodnot, převedena z datového typu float zpátky na datový typ int. Pro UDP a ICMP (ICMP6) provoz je postup stejný, akorát ze souboru vybíráme UDP nebo ICMP (ICMP6) provoz.

```
#!/bin/bash

path=/var/www/html/netflow/

while true; do

cd /home/student/flowsUpstreamIPV4
ls -1 | grep nfcapd > soubory.txt
tail -n 2 soubory.txt > soubory_dva.txt
head -1 soubory_dva.txt > soubor_vysl.txt


pocetBytuUpstreamIPV4TCP=0
nfdump -r `cat soubor_vysl.txt` -o "fmt:%pr%byt" | grep TCP | awk '{print $2}' > vypisTCP

for i in `cat vypisTCP`
do
    i=`echo "scale=3; $i / 1000.0" | bc`
    pocetBytuUpstreamIPV4TCP=`echo "scale=3; $i + $pocetBytuUpstreamIPV4TCP" | bc`
done

pocetBytuUpstreamIPV4TCP=${pocetBytuUpstreamIPV4TCP/.*}
```

Obrázek 4.2: *Zpracování přenesených bytů u TCP provozu*

Tento postup je totožný i pro sčítání datových toků a paketů. Jediný rozdíl je v tom, že hodnoty se pak nedělí 1024, ale přičítají se rovnou.

Po naplnění všech 36ti proměnných hodnotami musíme tyto hodnoty vložit do databáze. Nejdříve se přesuneme zpět do adresáře **/home/student**, kde máme uloženou databázi **flows.rrd** a následně si uložíme čas startu (v sekundách) do proměnné **start_hour** a **start_six** (počet sekund od 1.1. 1970 do teď mínus 3600, respektive 21600 - chceme vytvářet hodinový a šestihodinový graf). Následně vložíme do RRDTool databáze hodnoty proměnných [16].

```
cd /home/student/

start_hour=$((`date +%s`-3600))
start_six=$((`date +%s`-21600))

rrdtool update flows.rrd  N:`echo $pocetBytuUpstreamIPV4TCP`:`echo $pocetBytuUpstreamIPV4UDP`:`echo $pocetBytuUpstreamIPV4ICMP `:\
`echo $pocetBytuDownstreamIPV4TCP`:`echo $pocetBytuDownstreamIPV4UDP`:`echo $pocetBytuDownstreamIPV4ICMP `:\
`echo $pocetPaketuUpstreamIPV4TCP`:`echo $pocetPaketuUpstreamIPV4UDP`:`echo $pocetPaketuUpstreamIPV4ICMP `:\
`echo $pocetPaketuDownstreamIPV4TCP`:`echo $pocetPaketuDownstreamIPV4UDP`:`echo $pocetPaketuDownstreamIPV4ICMP `:\
`echo $pocetTokuUpstreamIPV4TCP`:`echo $pocetTokuUpstreamIPV4UDP`:`echo $pocetTokuUpstreamIPV4ICMP `:\
`echo $pocetTokuDownstreamIPV4TCP`:`echo $pocetTokuDownstreamIPV4UDP`:`echo $pocetTokuDownstreamIPV4ICMP `:\
`echo $pocetBytuUpstreamIPV6TCP`:`echo $pocetBytuUpstreamIPV6UDP`:`echo $pocetBytuUpstreamIPV6ICMP `:\
`echo $pocetBytuDownstreamIPV6TCP`:`echo $pocetBytuDownstreamIPV6UDP`:`echo $pocetBytuDownstreamIPV6ICMP `:\
`echo $pocetPaketuUpstreamIPV6TCP`:`echo $pocetPaketuUpstreamIPV6UDP`:`echo $pocetPaketuUpstreamIPV6ICMP `:\
`echo $pocetPaketuDownstreamIPV6TCP`:`echo $pocetPaketuDownstreamIPV6UDP`:`echo $pocetPaketuDownstreamIPV6ICMP `:\
`echo $pocetTokuUpstreamIPV6TCP`:`echo $pocetTokuUpstreamIPV6UDP`:`echo $pocetTokuUpstreamIPV6ICMP `:\
`echo $pocetTokuDownstreamIPV6TCP`:`echo $pocetTokuDownstreamIPV6UDP`:`echo $pocetTokuDownstreamIPV6ICMP `
```

Obrázek 4.3: *Vložení hodnot do RRDTool databáze*

Po vložení hodnot do RRDTool databáze můžeme přistoupit k vytváření samotných grafů. Obrázky s grafy se budou rovnou ukládat do adresáře **/var/www/html/NetFlow**. Popis dalších parametrů a atributů grafu:

- --title - titulek grafu
- --start, --end - časový začátek a konec grafu, zde je začátek o hodinu zpět
- --vertical-label - popis svislé osy
- -w, -h - parametry šířky a výšky grafu v pixelech
- DEF - definuje proměnné vyčtené z RRDTool databáze
- VDEF - vytváříme novou proměnnou, kterou pak využijeme u časového razítka
- LINE4 - vykreslí čáru tlustou 4 px v barvě, která je zadaná v hexadecimální soustavě
- GPRINT - vykreslí nám hodnoty dat pod graf (minimum, maximum, průměr) [16]

```
rrdtool graph `echo $path`byty1DownstreamIPV4.png --title "Hodinový provoz - downstream" \
--start $start_hour --end N --vertical-label "Počet přenesených kB" -w 500 -h 250 \
DEF:pocetBytuDoIPV4TCP=flows.rrd:pocetBytuDoIPV4TCP:AVERAGE \
DEF:pocetBytuDoIPV4UDP=flows.rrd:pocetBytuDoIPV4UDP:AVERAGE \
DEF:pocetBytuDoIPV4ICMP=flows.rrd:pocetBytuDoIPV4ICMP:AVERAGE \
VDEF:last=pocetBytuDoIPV4ICMP,LAST \
COMMENT:"------------------------------------------------------\\n" \
LINE4:pocetBytuDoIPV4TCP#FF0000:"TCP provoz" \
'GPRINT:pocetBytuDoIPV4TCP:MIN:Minimum\:%4.0lf' \
'GPRINT:pocetBytuDoIPV4TCP:MAX:Maximum\:%4.0lf' \
'GPRINT:pocetBytuDoIPV4TCP:AVERAGE:Průměr\:%4.0lf\j' \
COMMENT:"\\n" \
LINE4:pocetBytuDoIPV4UDP#00FF00:"UDP provoz" \
'GPRINT:pocetBytuDoIPV4UDP:MIN:Minimum\:%4.0lf' \
'GPRINT:pocetBytuDoIPV4UDP:MAX:Maximum\:%4.0lf' \
'GPRINT:pocetBytuDoIPV4UDP:AVERAGE:Průměr\:%4.0lf\j' \
COMMENT:"\\n" \
LINE4:pocetBytuDoIPV4ICMP#0000CD:"ICMP provoz" \
'GPRINT:pocetBytuDoIPV4ICMP:MIN:Minimum\:%4.0lf' \
'GPRINT:pocetBytuDoIPV4ICMP:MAX:Maximum\:%4.0lf' \
'GPRINT:pocetBytuDoIPV4ICMP:AVERAGE:Průměr\:%4.0lf\j' \
COMMENT:"\\n" \
COMMENT:"------------------------------------------------------\\n" \
GPRINT:last:"Poslední aktualizace\: %c":strftime
```

Obrázek 4.4: *Vykreslení hodinového grafu v RRDTool*

Pro šestihodinový graf je postup příkaz stejný. Pouze jako vstup atributu **--start** je vložena proměnná **start_six**, která reprezentuje čas v sekundách od 1.1. 1970, posunutý o 6 hodin zpět.

Sběr NetFlow dat na kolektoru

Před spuštěním samotného sběru datových toků, přijatých z exportéru, je nutno ještě upravit nastavení času, který je nutno posunout o 5 minut zpět. Program RRDTool totiž čte údaje o aktuálním čase z počítače a soubory se zachyceným NetFlow provozem jsou vytvářeny každých 5 minut a obsahují provoz, který proběhl před 0-5ti minutami. Pokud by jsme nechali nastavený čas tak, jak je, pak by vykreslované hodnoty neodpovídaly reálnému provozu, ale byly by vykreslené s 5ti-minutovým zpožděním.

Po nastavení času na kolektoru můžeme přistoupit k samotnému sběru NetFlow. K tomu použijeme nástroj nfcapd, který čte NetFlow data z určitého rozhraní a ukládá je do souborů. Jak již bylo zmíněno, soubory jsou opatřeny časovým razítkem YYYYMMddhhmm, např. **nfcapd.201703200845**.

Bude třeba si otevřít pět konzolí. Ve čtyřech konzolích bude spuštěn nástroj nfcapd pro upstream a downstream IPv4 nebo IPv6 provoz. Pomocí parametru -w se budou vytvářet každých 5 minut soubory se zachyceným provozem. Parametr **-p** definuje port, na kterém budeme datové toky přijímat, parametr **-n** definuje, název zařízení, ze kterého se budou datové toky sbírat, IP adresu rozhraní zdroje a do které složky se budou ukládat. Příkazem **./NetFlowDiplomka.sh** spustíme příslušný skript.

```
nfcapd -w -p 5534 \
-n CR3,192.168.57.1,/home/student/flowsUpstreamIPV4

nfcapd -w -p 5536 \
-n CR3,192.168.57.1,/home/student/flowsUpstreamIPV6

nfcapd -w -p 5564 \
-n CR6,192.168.59.1,/home/student/flowsUpstreamIPV4

nfcapd -w -p 5566 \
-n CR6,192.168.59.1,/home/student/flowsUpstreamIPV6


./NetFlowDIPlomka.sh
```

4.1.6. Instalace a konfigurace na PC6

Tento počítač bude mít za úkol generovat testovací provoz do sítě. K tomu bude třeba nainstalovat různé generátory provozu. Budeme používat nástroje **Mgen** (verze 5.02), **Mz** (verze 0.40)a **Hping3** (verze 3.0.0).

```
apt-get update
apt-get install mgen
apt-get install mz
```

```
apt-get install hping3
```

Po nainstalování těchto programů je ještě třeba nastavit adresy na rozhraní eth0. Tento postup byl již popsát v kapitole 3.2.4. Pro tento počítač se nastaví IPv4 adresa 192.168.57.2/24, IPv6 adresa 2001:db8:1001:57::2/64 a výchozí brána 192.168.57.1 (směrovač CR3) [17] [18] [19].

Nyní můžeme přistoupit k samotnému generování provozu. K tomu bude vhodné používat skripty, které budou automaticky spouštěny pomocí nástroje Cron.

Generování provozu pomocí Mgen

Nástroj Mgen používá ke generování provozu textové soubory, které obsahují sekvence příkazů, naplánované události popisující schéma generování provozu, čísla zdrojových / cílových portů, IP adres a velikost přenášených dat. Jeho výhodou je, že umí generovat i IPv6 provoz.

Obrázek níže ukazuje skript (název skriptu je **mgenIPv4**) pro generování TCP, UDP provozu nad protokolem IPv4. Jednotlivé části skriptu jsou popsány následovně:

- 0.0 - doba spuštění / ukončení dané události v sekundách
- ON, OFF - spuštění nebo zastavení dané události
- 1, 2, 3.. - číslo dané události
- TCP, UDP - typ provozu na 4. vrstě
- SRC - zdrojový port
- DST - cílová IP adresa s portem
- PERIODIC, POISSON, BURST - pakety mohou být vysílány pravidelně v určitých intervalech nebo ve staticky měnících se intervalech nebo shlukovitě [17]

```
0.0 ON 1 TCP SRC 21 DST 192.168.59.3/23 PERIODIC [2 128]
10.0 OFF 1
11.0 ON 2 TCP SRC 20 DST 192.168.59.3/20 PERIODIC [10 1024]
13.0 ON 3 UDP SRC 53 DST 192.168.59.2/53 PERIODIC [5 256]
15.0 OFF 3
16.0 ON 4 TCP SRC 443 DST 192.168.59.3/443 BURST [RANDOM 15 PERIODIC [20 1024] EXP 3]
25.0 OFF 2
46.0 OFF 4
```

Obrázek 4.5: *Skript v nástroji MGEN pro generování IPv4 provozu*

Skript výše tedy generuje prvních 10 sekund TCP provoz ze zdrojového portu 21 na cílovou IP adresu 192.168.59.3 na port 23 pravidelných intervalech s frekvencí 2 paketů za sekundu. Každý paket má velikost 128 bytů. Od 11. do 25. sekundy vysílá deset 1024 bytových paketů za sekundu na IP adresu 192.168.59.3 (PC7) z portu 20 na cílový port 20. Od 13. do 15. sekundy vysílá UDP provoz na port 53 a cílovou adresu 192.168.59.2 (PC5_Collector). Poslední 4. událost se spouští v 16. sekundě, kdy je generován TCP provoz na portu 443 s cílovou IP adresou 192.168.59.3 (PC7). Tento typ provozu má shlukovitý charakter - je vysíláno 20 paketů za sekundu, každý paket má velikost 1024 bytů v 15 sekundových intervalech. Délka trvání shlukovitého provozu jsou 3 sekundy.

Následující skript v nástroji Mgen generuje provoz nad protokolem IPv6. Tento skript je podobný tomu předchozímu, tudíž ho není třeba tak podrobně popisovat. Celkem byly vytvořeny ještě další 2 skripty (IPv4 a IPv6), které se lišily délkou trvání a typem generovaného provozu [17].

```
0.0 ON 1 TCP SRC 80 DST 2001:db8:1001:59::2/5000 PERIODIC [5 512]
5.0 ON 2 UDP SRC 53 DST 2001:db8:1001:59::2/53 PERIODIC [2 512]
7.0 OFF 1
8.0 OFF 2
9.0 ON 3 UDP SRC 53 DST 2001:db8:1001:57::3/53 POISSON [1 256]
10.0 ON 4 TCP SRC 443 DST 2001:db8:1001:59::3/443 BURST [RANDOM 5 POISSON [5 512] FIXED 3.0]
13.0 ON 5 TCP SRC 20 DST 2001:db8:1001:59::3/20 BURST [RANDOM 20 PERIODIC [10 1024] EXP 5.0]
20.0 OFF 3
25.0 OFF 4
30.0 ON 6 TCP SRC 32323 DST 2001:db8:1001:59::2/32323 POISSON [5 512]
40.0 OFF 6
50.0 OFF 5
```

Obrázek 4.6: *Skript v nástroji MGEN pro generování IPv6 provozu*

Generování provozu pomocí Mz (Mausezahn) a Hping3

Syntaxe příkazů u těchto programů je velmi podobná. Bohužel umožňují pouze generování IPv4 provozu. Níže je uveden skript (**mzIPv4_1.sh**)v jazyce Bash. První příkaz generuje z rozhraní eth0 na IP adresu 192.168.59.3 (PC7) TCP provoz s příznaky SYN od čísla portu 70 do 85. Počet odeslaných paketů na každý port je 5. Dále v cyklu while jsou každé 3 sekundy odesílány na adresu 192.168.57.1 (směrovač CR3) ICMP zprávy. Dále je z rozhraní eth0 na cílovou adresu 192.168.59.2 (PC5_Collector) vysílán UDP provoz z portu 20-24. Rovněž je vysílán i RTP provoz na cílovou adresu 192.168.59.3 (PC7) v počtu 20 paketů.

Nástroj Hping3 nám pak generuje TCP provoz na cílovou IP adresu 192.168.59.3 (PC7) na cílový port 80. Velikost paketu je 300 bytů a celkový počet odeslaných paketů je 20. Poslední příkaz nám generuje UDP provoz (parametr **-2**) také v počtu 20 paketů s velikostí jednoho paketu 500 bytů na portu 20 a na cílovou adresu 192.168.59.2 (PC5_Collector) [18][19].

```
#!/bin/bash
mz eth0 -B 192.168.59.3 -c 5 -t tcp "dp=70-85, flags=syn"

i=0
while [ $i -lt 10 ]; do
  mz -t icmp -B 192.168.57.1 -c 5
  i=$[$i+1]
  sleep 3
done

mz eth0 -B 192.168.59.2 -t udp "dp=20-24"
mz -t rtp id=11:11:33:22 -B 192.168.59.3 -c 20
hping3 -c 20 -d 300 -V -p 80 -s 5003 192.168.59.3
hping3 -2 -c 20 -d 500 -V -p 20 -s 20 192.168.59.2
```

Obrázek 4.7: *Skript pro generování provozu pomocí nástrojů Mz a Hping3*

Pro generování provozu pomocí nástroje Mz a Hping byl ještě vytvořen jeden skript s podobnými parametry. Ten bude uveden v příloze.

Zápis do souboru Crontab

Po vytvoření skriptů je nutno zajistit, aby se tyto skripty spouštěly automaticky v určitých časových intervalech. K tomu nám pomůže soubor **/etc/crontab**, do kterého zapíšeme časy a příkazy pro spouštění těchto skriptů. Jednotlivé skripty se budou spouštět každých 4, 5, 6, 8 nebo 12 minut. Po editaci souboru je vhodné restartovat službu Cron.

```
*/5 *    * * *   root    cd /home/student/ && mgen input mgenIPV4_1
*/8 *    * * *   root    cd /home/student/ && mgen input mgenIPV6
*/6 *    * * *   root    cd /home/student/ && mgen input mgenIPV4
*/4 *    * * *   root    cd /home/student/ && mgen input mgenIPV6_1
*/12 *    * * *   root    cd /home/student/ && ./mzIPV4_1.sh
```

Obrázek 4.8: *Výpis části souboru Crontab*

4.1.7. Instalace a konfigurace na PC7

Instalace a konfigurace na PC7 je totožná jako na PC6, proto zde budou uvedeny jen obrázky skriptů na generování provozu, které mírně odlišují v generování provozu a používají jiné cílové IP adresy.

Následující obrázky ukazují skripty, které generují IPv4 a IPv6 provoz. Cílovou stanicí je PC6 (192.168.57.2, respektive 2001:db8:1001:57::2) [17].

```
0.0 ON 1 TCP SRC 23 DST 192.168.57.2/23 POISSON [10 256]
10.0 OFF 1
13.0 ON 2 UDP SRC 53 DST 192.168.57.2/53 PERIODIC [3 128]
15.0 OFF 2
16.0 ON 3 TCP SRC 80 DST 192.168.57.2/80 POISSON [5 1024]
25.0 OFF 3
```

Obrázek 4.9: *Skript pro generování IPv4 provozu v nástroji MGEN*

```
0.0 ON 1 TCP SRC 80 DST 2001:db8:1001:57::2/5000 PERIODIC [5 512]
5.0 ON 2 UDP SRC 53 DST 2001:db8:1001:57::2/53 PERIODIC [2 512]
7.0 OFF 1
8.0 OFF 2
9.0 ON 3 UDP SRC 53 DST 2001:db8:1001:57::2/53 POISSON [1 256]
10.0 ON 4 TCP SRC 443 DST 2001:db8:1001:57::2/443 BURST [RANDOM 5 POISSON [5 512] FIXED 3.0]
13.0 ON 5 UDP SRC 20 DST 2001:db8:1001:57::2/20 BURST [RANDOM 20 PERIODIC [10 1024] EXP 5.0]
20.0 OFF 3
25.0 OFF 4
30.0 ON 6 TCP SRC 32323 DST 2001:db8:1001:57::2/32323 POISSON [10 512]
40.0 OFF 6
50.0 OFF 5
```

Obrázek 4.10: *Skript pro generování IPv6 provozu v nástroji MGEN*

Na následujícím obrázku 4.11 je skript v jazyce Bash, který generuje TCP, ICMP a UDP provoz nad protokolem IPv4 pomocí nástroje Hping3 [19].

```
#!/bin/bash

hping3 -c 20 -V -S -w 64 -p 80 -s 80 -d 400 192.168.57.2
hping3 -c 20 -V -F -w 64 -p 80 -s 80 -d 400 192.168.57.2
hping3 -1 -c 10 -d 200 192.168.57.1
hping3 -2 -c 10 -V -p 53 -s 53 -d 500 192.168.57.2
hping3 -2 -c 20 -V -p 5001  -d 1024 192.168.57.2
```

Obrázek 4.11: *Skript pro generování IPv4 provozu v nástroji Hping3*

Na posledním obrázku je generován provoz pomoci nástroje Mausezahn (Mz). Cílovou stanicí je opět PC7 [18].

```
#!/bin/bash
mz -t rtp id=11:12:41:11 -B 192.168.57.2 -c 20

i=0
while [ $i -lt 5 ]; do
  mz -t icmp -B 192.168.57.2 -c 2
  i=$[$i+1]
  sleep 2
done

mz eth0 -B 192.168.57.2 -t udp "dp=20-53"
```

Obrázek 4.12: *Skript pro generování IPv4 provozu v nástroji Mz*

4.1.8. Výpisy tabulek ze směrovačů

Následující výpis ze směrovačů CR3 a CR6 ukazuje nastavení exportu NetFlow dat na kolektor (PC5-Collector). Týká se protokolu IPv4. Je zde uvedena použitá verze NetFlow - verze 9, dále je zde IP adresa rozhraní kolektoru a port, na který se budou NetFlow data odesílat. Rovněž je zde uvedeno zdrojové rozhraní, ze kterého se budou datové toky posílat (FastEthernet0/0 s IP adresou 192.168.57.1, respektive 192.168.59.1). IP adresu z tohoto rozhraní je nutno nastavit pak na kolektoru jako adresu, ze které budou přicházet NetFlow datové toky.

```
CR3#sh ip flow export
Flow export v9 is enabled for main cache
  Exporting flows to 192.168.59.2 (5534)
  Exporting using source interface FastEthernet0/0
  Version 9 flow records
  4086 flows exported in 931 udp datagrams
```

Obrázek 4.13: *Výpis nastavení exportu pro IPv4 protokol na směrovači CR3*

```
CR6#sh ip flow export
Flow export v9 is enabled for main cache
  Exporting flows to 192.168.59.2 (5564)
  Exporting using source interface FastEthernet0/0
  Version 9 flow records
  6710 flows exported in 1051 udp datagrams
```

Obrázek 4.14: *Výpis nastavení exportupro IPv4 protokol na směrovači CR6*

Výpis informací o exportu IPv6 datových toků zobrazíme příkazem **show ipv6 flow export.** Lze si všimnout, že jako zdroj může být nastaveno buď rozhraní nebo rovnou IP adresa.

```
CR3#sh ipv6 flow export
Flow export v9 is enabled for main cache
  Exporting flows to 192.168.59.2 (5536)
  Exporting using source interface FastEthernet0/0
  Version 9 flow records
  4186 flows exported in 955 udp datagrams
```

Obrázek 4.15: *Výpis nastavení exportu pro IPv6 protokol na směrovači CR3*

```
CR6#sh ipv6 flow export
Flow export v9 is enabled for main cache
  Exporting flows to 192.168.59.2 (5566)
  Exporting using source IP address 192.168.59.1
  Version 9 flow records
  6717 flows exported in 1054 udp datagrams
```

Obrázek 4.16: *Výpis nastavení exportu pro IPv6 protokol na směrovači CR6*

Datové toky na směrovačů jsou ještě před odesláním na kolektor shromážděny do tzv. cache - tedy nějaké dočasné úložiště. Aktuální stav této paměti si lze vypsat příkazem **show ip cache flow.** Na obrázku 4.17 je výpis této paměti ze směrovače CR3. Obsahuje informace o aktivních datových tocích, neaktivních, nově navázaných a těch které budou exportovány na kolektor. Ve spodní části výpisu jsou pro každou službu informace o počtu přenesených paketů, bytů, délce trvání datového toku. Rovněž jsou zde i zdrojové a cílové IP adresy, mezi kterými probíhal nějaký provoz.

```
IP Flow Switching Cache, 278544 bytes
  4 active, 4092 inactive, 437 added
  9143 ager polls, 0 flow alloc failures
  Active flows timeout in 30 minutes
  Inactive flows timeout in 15 seconds
IP Sub Flow Cache, 25736 bytes
  4 active, 1020 inactive, 437 added, 437 added to flow
  0 alloc failures, 0 force free
  1 chunk, 1 chunk added
  last clearing of statistics never
Protocol         Total    Flows   Packets Bytes  Packets Active(Sec) Idle(Sec)
--------         Flows     /Sec     /Flow  /Pkt     /Sec     /Flow     /Flow
TCP-Telnet          52      0.0         1    53      0.0       0.5      10.4
TCP-FTP              5      0.0         1    40      0.0       0.0       1.4
TCP-FTPD            21      0.0         1    60      0.0       1.4      15.5
TCP-WWW             50      0.0         3   272      0.0       0.6       5.0
TCP-other          222      0.0         2    55      0.1       0.0       7.1
UDP-DNS             15      0.0        20   198      0.0       1.8      15.5
UDP-other           38      0.0        19   517      0.1      94.5      14.9
ICMP                30      0.0        17   170      0.1      11.6      15.5
Total:             433      0.0         5   257      0.4       9.3       9.1
```

Obrázek 4.17: *Výpis NetFlow cache pro IPv4 protokol na směrovači CR3*

```
SrcIf         SrcIPaddress    DstIf         DstIPaddress    Pr SrcP DstP  Pkts
Fa0/0         0.0.0.0         Null          255.255.255.255 11 0044 0043   139
Fa0/0         192.168.57.2    Fa0/1         192.168.59.3    01 0000 0303    12
Fa0/0         192.168.57.2    Fa0/1         192.168.59.3    06 0014 0014     1
Fa0/0         192.168.57.2    Fa0/1         192.168.59.3    06 0050 0057     1
```

Obrázek 4.18: *Výpis NetFlow cache pro IPv4 protokol na směrovači CR3*

Na CR6 směrovači je výpis podobný.

```
IP Flow Switching Cache, 278544 bytes
  1 active, 4095 inactive, 165 added
  5502 ager polls, 0 flow alloc failures
  Active flows timeout in 30 minutes
  Inactive flows timeout in 15 seconds
IP Sub Flow Cache, 25736 bytes
  1 active, 1023 inactive, 165 added, 165 added to flow
  0 alloc failures, 0 force free
  1 chunk, 1 chunk added
  last clearing of statistics never
Protocol         Total    Flows   Packets Bytes  Packets Active(Sec) Idle(Sec)
--------         Flows     /Sec     /Flow  /Pkt     /Sec     /Flow     /Flow
TCP-Telnet           6      0.0         1    53      0.0       0.0      10.8
TCP-FTP              4      0.0         1    40      0.0       0.0       1.2
TCP-FTPD             3      0.0         1    40      0.0       0.0       1.2
TCP-WWW              4      0.0        20   613      0.0       1.9       1.3
TCP-other           16      0.0         5    40      0.0       0.0       1.2
UDP-DNS              3      0.0         2   124      0.0       0.5      15.6
UDP-other           70      0.0         7   620      0.1      25.9      15.2
ICMP                58      0.0         2   342      0.0       4.4      15.2
Total:             164      0.0         5   506      0.2      12.6      12.8
```

Obrázek 4.19: *Výpis NetFlow cache pro IPv4 protokol na směrovači CR6*

Výpis IPv6 cache lze provést pomocí příkazu **show ipv6 flow cache.** Zde jsou rovněž vypsány aktivní datové toky, neaktivní a přidané, které se budou odesílat na kolektor. Lze si všimnout, že se ve spodní části výpisu již nevypisují informace o službách, ale pouze zdrojové a cílové IPv6 adresy, protokol a porty v hexadecimálním tvaru a počet přenesených paketů. Kromě globálních IPv6 adres, které jsou nastavené ručně na rozhraní, jsou zde vypsány i lokální IPv6 adresy, mezi kterými probíhá různá komunikace, např. přenos keepalive zprávy.

```
CR3#sh ipv6 flow cache
IP packet size distribution (3621 total packets):
   1-32   64   96  128  160  192  224  256  288  320  352  384  416  448  480
   .000 .059 .512 .001 .008 .000 .063 .000 .000 .000 .003 .000 .000 .000 .000

    512  544  576 1024 1536 2048 2560 3072 3584 4096 4608
   .000 .000 .018 .041 .292 .000 .000 .000 .000 .000 .000

IP Flow Switching Cache, 475168 bytes
  4 active, 4092 inactive, 458 added
  9147 ager polls, 0 flow alloc failures
  Active flows timeout in 30 minutes
  Inactive flows timeout in 15 seconds
IP Sub Flow Cache, 33928 bytes
  0 active, 1024 inactive, 0 added, 0 added to flow
  0 alloc failures, 0 force free
  1 chunk, 1 chunk added
SrcAddress        InpIf    DstAddress       OutIf    Prot SrcPrt DstPrt Packets
2001:DB...7C:2947 Fa0/0    FE80::2...C:4820 Local    0x3A 0x0000 0x8800 1
FE80::7...7C:2947 Fa0/0    FE80::2...C:4820 Local    0x3A 0x0000 0x8800 1
2001:DB...7C:2947 Fa0/0    2001:71...01::53 Local    0x11 0xD637 0x0035 16
FE80::7...7C:2947 Fa0/0    FE80::2...C:4820 Local    0x3A 0x0000 0x8700 1
```

Obrázek 4.20: *Výpis NetFlow cache pro IPv6 protokol na směrovači CR3*

Pomocí příkazu **show ipv6 interface brief** na směrovači CR3 si lze vypsat přiřazení určité globální a lokální IPv6 adresy na rozhraní a stav rozhraní. Lokální IPv6 adresy mohou být také použity ke komunikaci.

```
CR3#sh ipv6 int br
FastEthernet0/0            [up/up]
    FE80::21E:F7FF:FEAC:4820
    2001:DB8:1001:57::1
FastEthernet0/1            [up/up]
    FE80::21E:F7FF:FEAC:4821
    2001:DB8:1001:58::1
```

Obrázek 4.21*: Výpis IPv6 adres přiřazených na síťových rozhraních na směrovači CR3*

Výpis na směrovači CR6 je identický. Vyskytují se zde rovněž datové toky se zdrojovými a cílovými IPv6 adresami, které jsou nastavené na rozhraní ručně (2001:db8:1001:57::2, 2001:db8:1001:59::3).

```
CR6#sh ipv6 flow cache
IP packet size distribution (4082 total packets):
   1-32   64   96  128  160  192  224  256  288  320  352  384  416  448  480
   .000 .067 .470 .021 .002 .079 .000 .000 .000 .025 .000 .000 .000 .000 .000

    512  544  576 1024 1536 2048 2560 3072 3584 4096 4608
   .000 .000 .081 .016 .234 .000 .000 .000 .000 .000 .000

IP Flow Switching Cache, 475168 bytes
  14 active, 4082 inactive, 616 added
  11028 ager polls, 0 flow alloc failures
  Active flows timeout in 30 minutes
  Inactive flows timeout in 15 seconds
IP Sub Flow Cache, 33928 bytes
  0 active, 1024 inactive, 0 added, 0 added to flow
  0 alloc failures, 0 force free
  1 chunk, 1 chunk added
SrcAddress        InpIf    DstAddress       OutIf    Prot SrcPrt DstPrt Packets
2001:DB...7C:299C Fa0/0    2001:DB...:57::2 Fa0/1    0x11 0x0015 0x0015 1
2001:DB...7C:75AD Fa0/0    2001:71...01::53 Local    0x11 0x9036 0x0035 12
FE80::7...7C:299C Fa0/0    FE80::2...C:55F0 Local    0x3A 0x0000 0x8800 1
2001:DB...7C:299C Fa0/0    FE80::2...C:55F0 Local    0x3A 0x0000 0x8800 1
2001:DB...1:59::3 Fa0/0    FE80::2...C:55F0 Local    0x3A 0x0000 0x8800 1
2001:DB...1:59::2 Fa0/0    FE80::2...C:55F0 Local    0x3A 0x0000 0x8800 1
```

Obrázek 4.22*: Výpis NetFlow cache pro IPv6 protokol na směrovači CR6*

Pomocí příkazu **show IPv6 interface brief** na směrovači CR3 si lze taktéž vypsat přiřazení určité globální a lokální IPv6 adresy na rozhraní.

```
CR6#sh ipv6 int br
FastEthernet0/0            [up/up]
    FE80::21E:F7FF:FEAC:55F0
    2001:DB8:1001:59::1
FastEthernet0/1            [up/up]
    FE80::21E:F7FF:FEAC:55F1
    2001:DB8:1001:58::2
```

Obrázek 4.23: *Výpis IPv6 adres přiřazených na síťových rozhraních na směrovači CR6*

4.1.9. Výpisy z kolektoru (PC5-Collector)

Na kolektoru (PC5-Collector) jsou každých 5 minut vytvářeny soubory, obsahující zachycený provoz, který byl přijat z exportéru. Tyto soubory později zpracovává skript **NetFlow_diplomka.sh** a ze zpracovaných hodnot vytváří grafy. Některý ze souborů si lze také vypsat manuálně pomocí příkazu **nfdump** (**nfdump -r nfcapd.201703211625,** soubor z 21.3. 2017). Následující výpis na kolektoru (PC5-Collector) ukazuje informace o datových tocích, které byly vytvořeny mezi 16:20 a 16:25. Jedná se o IPv4 datové toky ve směru downstream. Celkem jich bylo víc, ale vypsal jsem jen prvních 10. Jsou zde informace o počátku datového toku, jeho trvání, použitý protokol, zdrojová a cílová IP adresa, počet přenesených paketů, počet přenesených bytu a počet datových toků - je tam vždy 1. Na konci je vypsán celkový počet přenesených bytů, paketů a průměrný počet přenesených dat (bytů, paketů) za sekundu pro vypsaný počet datových toků. Pomocí dalších atributů příkazu **nfdump** si lze vypsat i další informace (BGP autonomní systém, informace o MPLS značkách, zdrojové a cílové MAC adresy).

```
Date flow start          Duration Proto      Src IP Addr:Port          Dst IP Addr:Port   Packets    Bytes Flows
Skip unknown record type 7
PANIC! - Verify map id 0: ERROR: element id 27 out of range [25]!
2017-03-21 16:20:00.224     0.000 TCP       192.168.59.3:23     ->     192.168.57.2:21          1       40     1
2017-03-21 16:20:11.228     0.000 TCP       192.168.59.3:20     ->     192.168.57.2:20          1       40     1
2017-03-21 16:20:16.228     0.000 TCP       192.168.59.3:443    ->     192.168.57.2:443         1       40     1
2017-03-21 16:22:00.728     0.388 UDP       192.168.59.3:30000 ->      192.168.57.2:30000      20     4000     1
2017-03-21 16:22:01.248     8.492 ICMP      192.168.59.3:0      ->     192.168.57.2:0.0        10      280     1
2017-03-21 16:22:11.772     0.000 UDP       192.168.59.3:0      ->     192.168.57.2:46          1       28     1
2017-03-21 16:22:11.772     0.000 UDP       192.168.59.3:0      ->     192.168.57.2:47          1       28     1
2017-03-21 16:22:11.772     0.000 UDP       192.168.59.3:0      ->     192.168.57.2:48          1       28     1
2017-03-21 16:22:11.772     0.000 UDP       192.168.59.3:0      ->     192.168.57.2:49          1       28     1
2017-03-21 16:22:11.772     0.000 UDP       192.168.59.3:0      ->     192.168.57.2:50          1       28     1
Summary: total flows: 10, total bytes: 4540, total packets: 38, avg bps: 276, avg pps: 0, avg bpp: 119
Time window: 2017-03-21 16:20:00 - 2017-03-21 16:24:38
Total flows processed: 102, Blocks skipped: 0, Bytes read: 8636
Sys: 0.000s flows/second: 0.0        Wall: 0.001s flows/second: 92559.0
```

Obrázek 4.24: *Výpis IPv4 datových toků na kolektoru (PC5_Collector)*

Pro IPv6 datové toky je výpis podobný. Zde se ovšem vyskytují IPv6 adresy. Lze si všimnout, že se zde vyskytují i jiné IPv6 adresy se stejným prefixem, jako byly nakonfigurovány na jednotlivé počítače. Jedná se o globální IPv6 adresy, které byly na jednotlivá rozhraní přidány automaticky. Přiřazení jednotlivých IPv6 adres si lze zobrazit pomocí příkazu **ifconfig**. Opět je zde část IPv6 adresy useknutá. Pro zobrazení celé IPv6 adresy je nutno přidat do příkazu **nfdump** parametr **-o line6.**

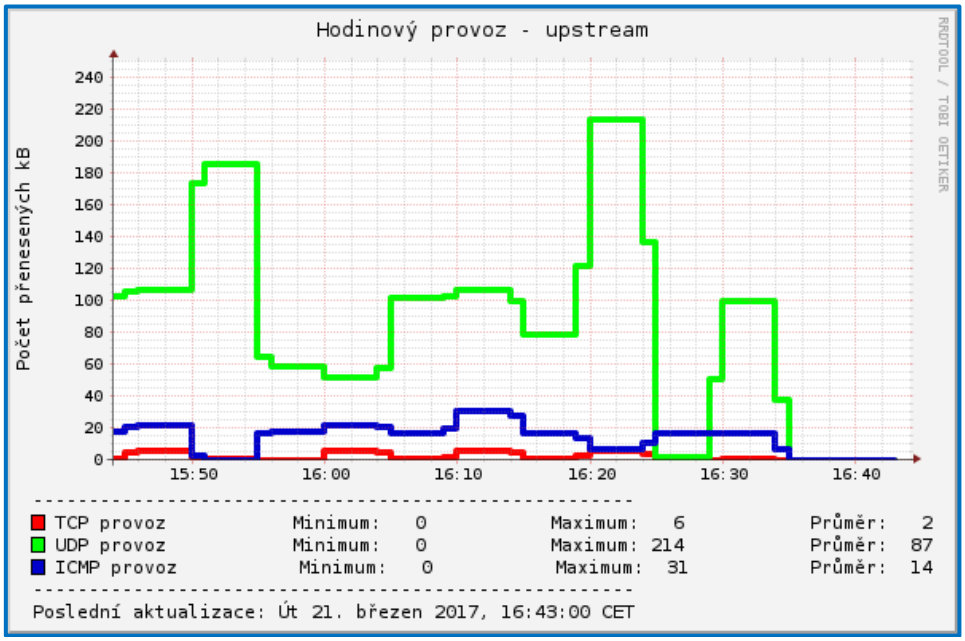
```
Date flow start          Duration Proto      Src IP Addr:Port          Dst IP Addr:Port   Packets    Bytes Flows
Skip unknown record type 7
PANIC! - Verify map id 0: ERROR: element id 27 out of range [25]!
2017-03-21 16:20:01.008     0.000 TCP   2001:db..1:57::2.5000  -> 2001:db..7c:299c.80          1       60     1
2017-03-21 16:20:11.012     0.000 TCP   2001:db..1:57::2.443   -> 2001:db..7c:299c.443         1       60     1
2017-03-21 16:20:00.636     0.000 TCP   2001:db..7c:2947.80    -> 2001:db..1:59::2.5000        1       80     1
2017-03-21 16:20:05.640     0.000 ICMP6 2001:db..7c:2947.0     -> fe80::2..ac:4820.136.0       1       64     1
2017-03-21 16:20:05.648     0.000 ICMP6 fe80::7..7c:2947.0     -> fe80::2..ac:4820.135.0       1       72     1
2017-03-21 16:20:06.184     0.000 ICMP6 2001:db..1:57::2.0     -> fe80::2..ac:4820.136.0       1       64     1
2017-03-21 16:20:05.640     2.500 UDP   2001:db..7c:2947.53    -> 2001:db..1:59::2.53          6     3360     1
2017-03-21 16:20:10.640     0.000 TCP   2001:db..7c:2947.443   -> 2001:db..1:59::3.443         1       80     1
2017-03-21 16:20:10.648     0.000 ICMP6 fe80::7..7c:2947.0     -> fe80::2..ac:4820.136.0       1       64     1
2017-03-21 16:20:13.640     0.000 TCP   2001:db..7c:2947.20    -> 2001:db..1:59::3.20          1       80     1
Summary: total flows: 10, total bytes: 3984, total packets: 15, avg bps: 2450, avg pps: 1, avg bpp: 265
Time window: 2017-03-21 16:20:00 - 2017-03-21 16:24:35
Total flows processed: 49, Blocks skipped: 0, Bytes read: 5752
Sys: 0.000s flows/second: 0.0        Wall: 0.001s flows/second: 41350.2
```

Obrázek 4.25*: Výpis IPv6 datových toků na kolektoru (PC5_Collector)*

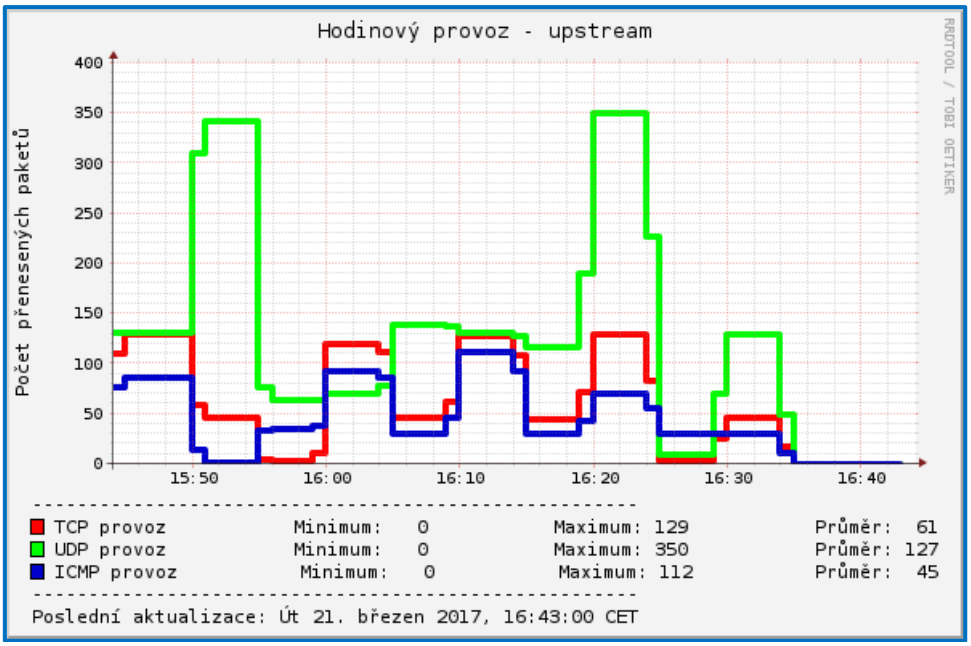
4.1.10. Grafy

Výstupem skriptu je celkem 24 grafů pro počet přenesených kB, paketů a datových toků. Ke každému protokolu (IPv4, IPv6) a směru provozu (downstream, upstream) byl vždy vytvořen jeden hodinový a jeden šestihodinový graf. Grafy se zobrazovaly na webové stránce, která se automaticky po minutě aktualizovala, aby zobrazovala aktuální obrázky grafů.

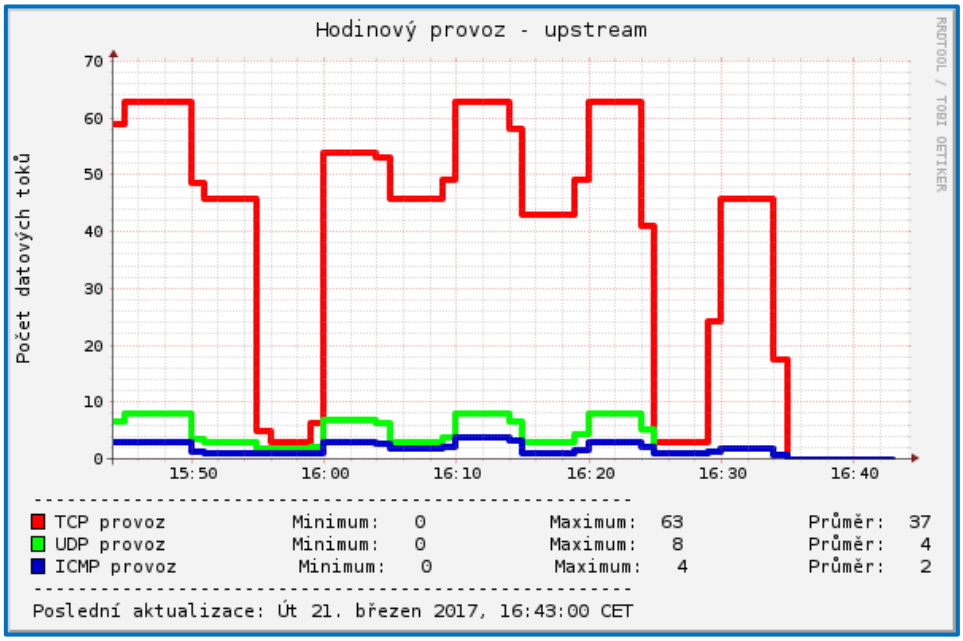
Každý graf obsahuje nadpis, ve kterém je uvedeno, jak je dlouhý (hodinový, šestihodinový) a směr provozu. Na svislé ose je popis, co se monitoruje. V každém grafu se vykresluje TCP, UDP, ICMP provoz a jejich minimální hodnota, maximální hodnota a průměrná hodnota. Na konci grafu je časové razítko, které informuje o poslední aktualizaci grafu [16].



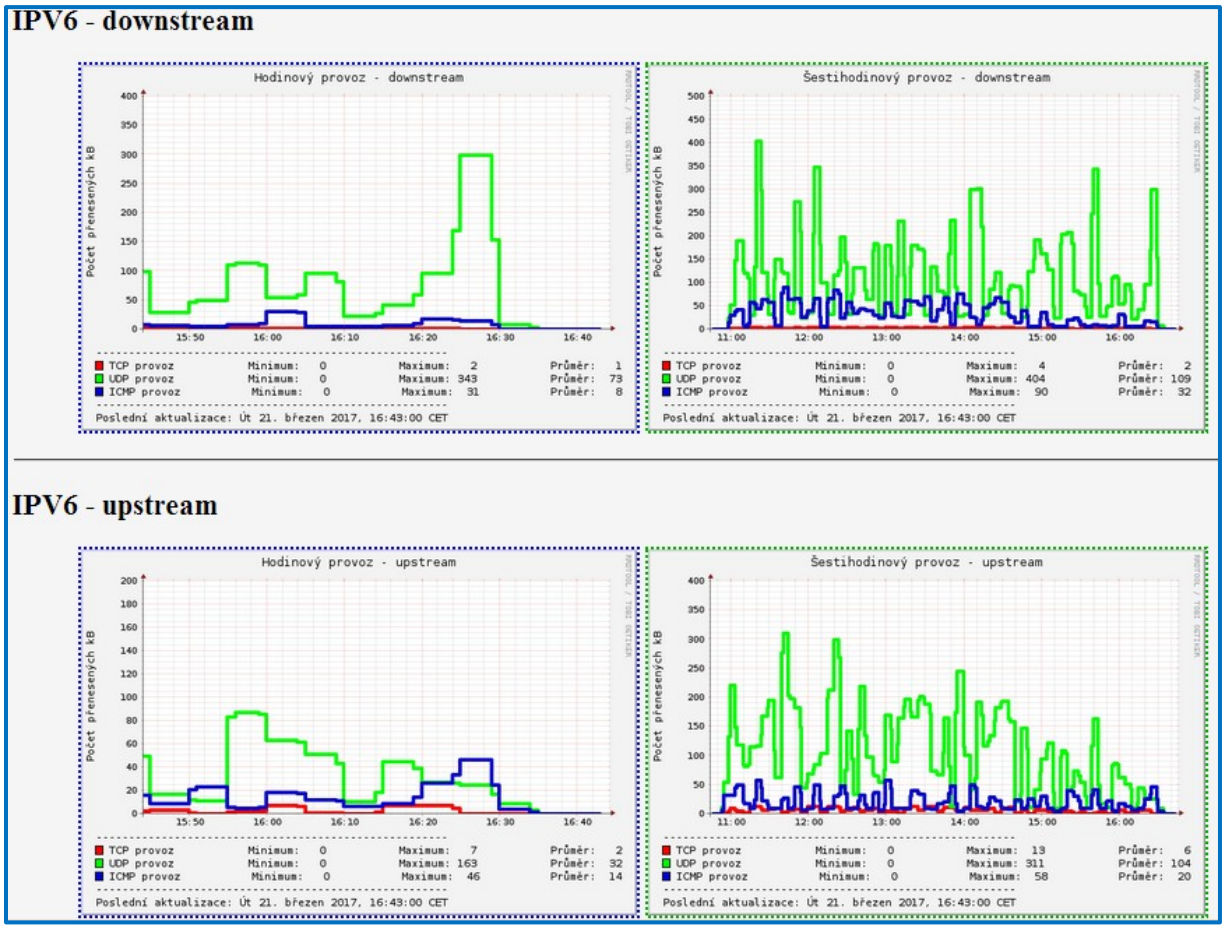
Obrázek 4.26: *Hodinový graf zobrazující počet přenesených kB pro směr upstream (IPv4)*



Obrázek 4.27: *Hodinový graf zobrazující počet přenesených paketů pro směr upstream (IPv4)*



Obrázek 4.28: *Hodinový graf zobrazující počet datových toků pro směr upstream (IPv4)*



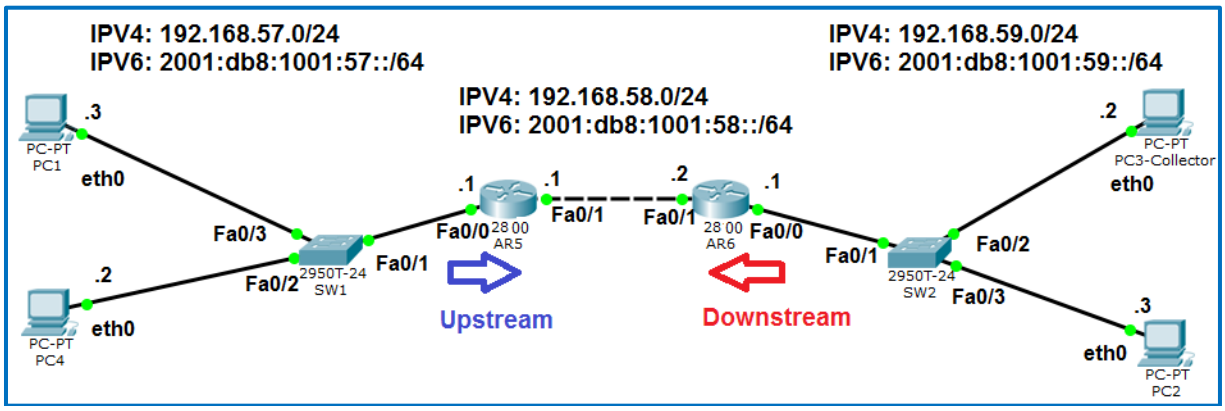
Obrázek 4.29: *Grafy zobrazující počet přenesených kB (IPv6)*

4.2. Konfigurace Flexibilního NetFlow na Cisco 2800 pro IPv6

Jak již bylo zmíněno, pro sběr IPv6 provozu se zde používá Flexibiblní NetFlow. Pro sběr informací o datových tocích, které vstupují do rozhraní, je zde použita předdefinovaná šablona, tudíž co se týče funkčnosti, je toto nastavení identické, jako u klasického NetFlow. Pro sběr IPv4 datových toků je zde použita konfigurace jako u klasického NetFlow, který tento směrovač také podporuje.

4.2.1. Topologie

Topologie je téměř identická, jako v předchozí kapitole. Pouze k přepínači SW1 bylo připojen další počítač (PC1) a názvy počítačů a Cisco směrovačů se změnily. Celkem se tedy topologie skládá ze 4 počítačů s operačním systémem Ubuntu 14.04. Všechny počítače měly jedno síťové rozhraní, pomocí kterého byly připojeny do Cisco přepínačů. Jeden z počítačů má funkci kolektoru (PC3_Collector). Dále se topologie skládá ze dvou Cisco směrovačů 2800 s verzí IOSu 12.4(22)T. Na těchto směrovačích je nakonfigurován export datových toků pro každý směr provozu zvlášť (směrovač AR5 analyzuje data ve směru upstream, směrovač AR6 ve směru downstream) a také pro oba protokoly současně (IPv4 a IPv6). Aby bylo možné sbírat informace o provozu v obou směrech zvlášť, bylo nutno použít v topologii 2 směrovače -při použití klasického NetFlow protokolu by sice bylo možné sbírat datové toky z obou směrů, avšak nebylo by možné je pak oddělit na kolektoru. Při použití Flexibilního NetFlow stačí použít jeden směrovač, viz. kapitola 4.3. Topologie sítě s adresováním jednotlivých stanic je uvedena na obrázku níže [12][14][15].



Obrázek 4.30: *Topologie sítě*

4.2.2. Konfigurace na směrovači AR5

Nejdříve je nutno si pojmenovat směrovač (v našem případě AR3) a nakonfigurovat přeposílání IPv6 unicastových paketů. Aby bylo možno aktivovat sběr IPv6 provozu na rozhraních, je nutno také aktivovat CEF (Cisco Express Forwarding) pro IPv6.

```
Router(config)#hostname AR5
AR5(config)#ipv6 unicast-routing
AR5(config-if)#ipv6 cef
AR5#clock set 12:00:00 21 Mar 2017
```

Dále je nutno nakonfigurovat IPv4 a IPv6 adresy na jednotlivé rozhraní a aktivovat tato

rozhraní. Uvedu zde konfiguraci adres pro jedno rozhraní směrovače, pro ostatní rozhraní je konfigurace odlišná pouze v adresách.

```
AR5(config)#interface FastEthernet 0/0
AR5(config-if)#ip address 192.168.57.1 255.255.255.0
AR5(config-if)#ip address 2001:db8:1001:57::1/64
AR5(config-if)#no shutdown
```

Po nakonfigurování adres aktivujeme směrovací algoritmus OSPF pro IPv4 a OSPVv3 pro IPv6 na jednotlivých rozhraních. Pro směrovač AR5 jsem zvolil router-id 1.1.1.1.

```
AR5(config)#router ospf 1
AR5(config-router)#network 192.168.57.0 0.0.0.255 area 0
AR5(config-router)#network 192.168.58.0 0.0.0.255 area 0

AR5(config)#ipv6 router ospf 1
AR5(config-router)#router-id 1.1.1.1

AR5(config)#interface FastEthernet 0/0
AR5(config-if)#IPv6 ospf 1 area 0

AR5(config)#interface FastEthernet 0/1
AR5(config-if)#IPv6 ospf 1 area 0
```

Nyní je nutno aktivovat NetFlow pro sběr IPv4 datových toků a Flexibilní NetFlow pro sběr IPv6 datových toků. Nejdříve bude popsána konfigurace pro sběr IPv4 datových toků, která se nijak neliší od předchozí konfigurace na CRx směrovačích.

Stejně jak v předchozím případě, pro sběr datových toků pro IPv4 protokol je nutno nakonfigurovat IP adresu a port kolektoru. V našem případě IP adresa kolektoru bude 192.168.58.2 a port 5554. Zdrojové rozhraní směrovače je nastaveno na FastEthernet0/0. Pak nastavíme verzi NetFlow protokolu- v našem případě verze . A nakonec je třeba aktivovat na rozhraní sběr datových toků. Rovněž je důležité nastavit směr - je možný směr provozu do rozhraní (ingress) nebo ven z rozhraní (egress) nebo oba současně. Zvolil jsem směr do rozhraní (ingress), protože je žádoucí mít přehled i o provozu, který ještě nebyl směrovačem vyfiltrován. Tudíž tento směrovač bude zobrazovat grafy pro upstream provoz.

```
AR5(config)#ip flow-export destination 192.168.59.2 5554
AR5(config)#ip flow-export source FastEthernet 0/0
AR5(config)#ip flow-export version 9

AR5(config)#interface FastEthernet 0/0
AR5(config-if)#ip flow ingress
```

Konfigurace Flexibilního NetFlow pro sběr IPv6 datových toků je o něco složitější. Skládá se z konfigurace tzv. flow exportéru a flow monitoru. V kapitole 4.3. je popsána konfigurace i sekce flow record - zde vybíráme podle jakých kriterií chceme provoz analyzovat a jaké parametry se mají sbírat.

Nejdříve je nutno nakonfigurovat flow exportér. Název flow exportéru jsem zvolil "**IPV6**", ale je možno si zvolit libovolný název. V této sekci je nutno nakonfigurovat cílovou IP adresu, kam se budou NetFlow data posílat - v našem případě 192.168.59.2 na portu 5556. Jako zdrojové rozhraní bylo zvoleno FastEthernet0/0.

Pro flow monitor byl také zvolen název "**IPV6**". V této části konfigurace je potřeba definovat záznam (record). Byl použit příkaz "**record NetFlow ipv6 original-input**", který emuluje funkčnost klasického NetFlow pro vstupní provoz IPv6 protokolu. Ve výpisu směrovaču AR5 si lze pak všimnout, podle který polí se vybírá provoz pro analýzu. Nyní je třeba ještě přiřadit flow monitoru správný flow exportér, v našem případě "**IPV6**".

Nakonec přiřadíme flow monitor ke správnému rozhraní a tímto se začíná analyzovat provoz.

```
AR5(config)#flow exporter IPV6
AR5(config-flow-exporter)#destination 192.168.59.2
AR5(config-flow-exporter)#source FastEthernet0/0
AR5(config-flow-exporter)#transport udp 5556

AR5(config)#flow monitor IPV6
AR5(config-flow-monitor)#record netflow ipv6 original-input
AR5(config-flow-monitor)# exporter IPV6

AR5(config)#interface FastEthernet 0/0
AR5(config-if)#IPv6 flow monitor IPv6 input
```

4.2.3. Konfigurace na směrovači AR6

Konfigurace na směrovači AR6 je velmi podobná, jako na AR5. Opět je nutno si pojmenovat směrovač (v našem případě AR6) a nakonfigurovat přeposílání IPv6 unicastových paketů. Aby bylo možno aktivovat sběr IPv6 provozu na rozhraních, je nutno také aktivovat CEF (Cisco Express Forwarding) pro IPv6. Rovněž je nutné nastavit správný čas na směrovači.

```
Router(config)#hostname AR6
AR6(config)#ipv6 unicast-routing
AR6(config-if)#ipv6 cef
AR6#clock set 12:00:00 21 Mar 2017
```

Dále je nutno nakonfigurovat IPv4 a IPv6 adresy na jednotlivé rozhraní a aktivovat tato rozhraní. Uvedu zde konfiguraci adres pro jedno rozhraní směrovače, pro ostatní rozhraní je konfigurace odlišná pouze v adresách.

```
AR6(config)#interface FastEthernet 0/0
AR6(config-if)#ip address 192.168.59.1 255.255.255.0
AR6(config-if)#ip address 2001:db8:1001:59::1/64
AR6(config-if)#no shutdown
```

Po nakonfigurování adres aktivujeme směrovací algoritmus OSPF pro IPv4 a OSPVv3 pro IPv6. Pro směrovač AR6 jsem zvolil router-id 2.2.2.2.

```
AR6(config)#router ospf 1
AR6(config-router)#network 192.168.58.0 0.0.0.255 area 0
AR6(config-router)#network 192.168.59.0 0.0.0.255 area 0

AR6(config)#ipv6 router ospf 1
AR6(config-router)#router-id 2.2.2.2

AR6(config)#interface FastEthernet 0/0
AR6(config-if)#ipv6 ospf 1 area 0

AR6(config)#interface FastEthernet 0/1
AR6(config-if)#ipv6 ospf 1 area 0
```

Konfigurace Flexibilního NetFlow na AR6 je opět téměř identická. Liší se pouze v jiném portu, na kterém bude kolektor přijímat NetFlow data. Jedná se o port 5566.

```
AR5(config)#flow exporter IPV6
AR5(config-flow-exporter)#destination 192.168.59.2
AR5(config-flow-exporter)#source FastEthernet0/0
AR5(config-flow-exporter)#transport udp 5566

AR5(config)#flow monitor IPV6
AR5(config-flow-monitor)#record NetFlow ipv6 original-input
AR5(config-flow-monitor)# exporter IPV6

AR5(config)#interface FastEthernet 0/0
AR5(config-if)#ipv6 flow monitor IPV6 input
```

4.2.4. Konfigurace na počítačích

Co se týče konfigurace na počítačích, tak ta je stejná, jako v předchozí kapitole. Pouze skripty pro generování provozu se spouštějí v jiných časových úsecích.

Na kolektoru se rovněž používá stejná databáze, struktura složek a skript pro zpracovávání přijatých NetFlow dat, tudíž není třeba znovu popisovat postup.

4.2.5. Výpisy tabulek ze směrovačů

Následující výpis ze směrovačů AR5 a AR6 ukazuje nastavení exportu NetFlow dat na kolektor (PC3-Collector). Týká se protokolu IPv4. Je zde uvedena použitá verze NetFlow - verze 9, dále je zde IP adresa rozhraní kolektoru a port, na který se budou NetFlow data odesílat. Rovněž je zde uvedena IP adresa a rozhraní, ze kterého se budou datové toky posílat (FastEthernet0/0 s IP adresou 192.168.57.1, respektive u AR6 IP adresa 192.168.59.1). IP adresu z tohoto rozhraní je nutno nastavit pak na kolektoru jako adresu, ze které budou přicházet NetFlow datové toky. Oproti předchozí konfiguraci na CRx směrovačích s IOSem 12.3(14)YT, zde přibyla zdrojová IP adresa.

```
AR5#show ip flow export
Flow export v9 is enabled for main cache
  Export source and destination details :
  VRF ID : Default
    Source(1)       192.168.57.1 (FastEthernet0/0)
    Destination(1)  192.168.59.2 (5554)
  Version 9 flow records
  299 flows exported in 49 udp datagrams
```

Obrázek 4.31: *Výpis nastavení exportu pro IPv4 protokol na směrovači AR5*

```
AR6#sh ip flow export
Flow export v9 is enabled for main cache
  Export source and destination details :
  VRF ID : Default
    Source(1)       192.168.59.1 (FastEthernet0/0)
    Destination(1)  192.168.59.2 (5564)
  Version 9 flow records
  765 flows exported in 230 udp datagrams
```

Obrázek 4.32: *Výpis nastavení exportu pro IPv4 protokol na směrovači AR6*

Výpisy dalších tabulek pro IPv4 protokol jsou podobné, jako u konfigurace CR směrovačů. Budou proto uvedeny v příloze.

Výpisy tabulek Flexibilního NetFlow jsou oproti klasickému NetFlow odlišné.

Pro výpis informací o konfiguraci všech flow monitorů na směrovači AR5 použijeme příkaz **show flow monitor**. Na každém směrovači byl nakonfigurován pouze jeden monitor pro sběr IPv6 datových toků. Výpis je stejný i pro směrovač AR6. Jsou zde vypsány použité záznamy, exportéry, velikost cache atd.

```
AR5#show flow monitor
Flow Monitor IPV6:
  Description:       User defined
  Flow Record:       netflow ipv6 original-input
  Flow Exporter:     IPV6
  Cache:
    Type:              normal
    Status:            allocated
    Size:              4096 entries / 540704 bytes
    Inactive Timeout:  15 secs
    Active Timeout:    1800 secs
    Update Timeout:    1800 secs
```

Obrázek 4.33: *Výpis nastavení pro flow monitor IPv6 na směrovači AR5*

Pro výpis informací o záznamech, na základě kterých dochází k analýze a sběru provozu použijeme příkaz **show flow record NetFlow IPv6 original-input.** Jak již bylo řečeno, byla zvolena

defaultní šablona k analýze a sběru IPv6 provozu. Z výpisu je patrné, že daný datový tok musí mít například IPv6 zdrojovou a cílovou adresu, musí se jednat o IPv6 protokol nebo se musí jednat o provoz, který vstupuje do rozhraní. Jedná se o tzv. klíčová pole (příkaz **match**). Pokud daný paket splňuje tyto požadavky, jsou některá pole z něj vyčteny a je vytvořen záznam ve Flexibilní NetFlow cache (příkaz **collect**). Z datového toku se vybírá časové razítko začátku a konce, počet přenesených bytů, paketů, zdrojová a cílová IPv6 adresa a maska, TCP příznaky, výstupní rozhraní atd. Pro směrovač AR6 je výpis identicky, bude zde tedy uveden pouze pro směrovač AR5.

```
AR5#show flow record netflow ipv6 original-input
flow record netflow ipv6 original-input:
  Description:        Traditional IPv6 input NetFlow with ASs
  No. of users:       1
  Total field space:  97 bytes
  Fields:
    match ipv6 traffic-class
    match ipv6 flow-label
    match ipv6 protocol
    match ipv6 extension map
    match ipv6 source address
    match ipv6 destination address
    match transport source-port
    match transport destination-port
    match interface input
    match flow direction
    match flow sampler
    collect routing source as
    collect routing destination as
    collect routing next-hop address ipv6
    collect ipv6 source mask
    collect ipv6 destination mask
    collect transport tcp flags
    collect interface output
    collect counter bytes
    collect counter packets
    collect timestamp sys-uptime first
    collect timestamp sys-uptime last
```

Obrázek 4.34: *Výpis nastavení pro flow record na směrovači AR5*

Nakonec si můžeme vypsat část samotné cache pro flow monitor IPv6. To provedeme pomocí příkazu show **flow monitor IPv6 cache**, kde "IPv6" je název monitoru, který jsem si definovali při konfiguraci. Na začátku výpisu jsou opět informace o typu a velikosti cache (4096 záznamů), nebo kolik záznamů obsahuje aktuálně. Jeden záznam se zde rovná jednomu datovému toku.

Velkou výhodou Flexibilního NetFlow je detailnější výpis informací o každém datovém toku. Konkrétně na obrázku 4.35 se jedná o datový tok z PC4 na PC3_Collector na portu 80. Oproti klasickému NetFlow zde přibyly například časové razítka, směr toku, výstupní rozhraní, ToS apod.

```
AR5#show flow monitor IPV6 cache
  Cache type:                               Normal
  Cache size:                                 4096
  Current entries:                              11
  High Watermark:                               24

  Flows added:                                 551
  Flows aged:                                  540
    - Active timeout      (  1800 secs)          0
    - Inactive timeout (    15 secs)           540
    - Event aged                                 0
    - Watermark aged                             0
    - Emergency aged                             0
```

Obrázek 4.35: *Výpis cache pro flow monitor IPv6 na směrovači AR5 - první část*

```
IPV6 SOURCE ADDRESS:       2001:DB8:1001:57:76D4:35FF:FE7C:29E6
IPV6 DESTINATION ADDRESS:  2001:DB8:1001:59::2
TRNS SOURCE PORT:          80
TRNS DESTINATION PORT:     80
INTERFACE INPUT:           Fa0/0
FLOW DIRECTION:            Input
FLOW SAMPLER ID:           0
IP PROTOCOL:               6
IP TOS:                    0x00
ip source as:              0
ip destination as:         0
ipv6 next hop address:     FE80::21E:F7FF:FEAC:4A63
ipv6 source mask:          /64
ipv6 destination mask:     /64
tcp flags:                 0x1E
interface output:          Fa0/1
counter bytes:             6136
counter packets:           14
timestamp first:           15:06:00.315
timestamp last:            15:06:01.315
```

Obrázek 4.36: *Výpis cache pro flow monitor IPv6 na směrovači AR5- druhá část*

Globální IPv6 adresu, která byla nakonfigurována automaticky, lze zjistit příkazem **ifconfig**. Tato IPv6 adresa se objevuje ve výpisu na obrázku 4.36 a patří PC4.

```
root@PC4:/home/student# ifconfig
eth0      Link encap:Ethernet  HWaddr 74:d4:35:7c:29:e6
          inet addr:192.168.57.2  Bcast:0.0.0.0  Mask:255.255.255.0
          inet6 addr: 2001:db8:1001:57::2/64 Scope:Global
          inet6 addr: fe80::76d4:35ff:fe7c:29e6/64 Scope:Link
          inet6 addr: 2001:db8:1001:57:76d4:35ff:fe7c:29e6/64 Scope:Global
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:33654 errors:0 dropped:57 overruns:0 frame:0
          TX packets:21982 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:14062772 (14.0 MB)  TX bytes:7463522 (7.4 MB)
```

Obrázek 4.37: *Výpis nastavení síťového rozhraní eth0 na PC4*

Na směrovači AR6 je výpis flow monitor cache tabulky podobný. Jsou zde však jiné zdrojové a cílové IPv6 adresy - adresy s jiným prefixem. Na obrázku 4.38 se jedná o jeden z několika datových toků mezi PC2 a PC4 na portu 443.

```
AR6#show flow monitor IPV6 cache
  Cache type:                               Normal
  Cache size:                                 4096
  Current entries:                               5
  High Watermark:                               25

  Flows added:                                 423
  Flows aged:                                  418
    - Active timeout      (  1800 secs)          0
    - Inactive timeout    (    15 secs)        418
    - Event aged                                 0
    - Watermark aged                             0
    - Emergency aged                             0
```

Obrázek 4.38: *Výpis cache pro flow monitor IPv6 na směrovači AR6 - první část*

```
IPV6 SOURCE ADDRESS:        2001:DB8:1001:59::3
IPV6 DESTINATION ADDRESS:   2001:DB8:1001:57:76D4:35FF:FE7C:29E6
TRNS SOURCE PORT:           443
TRNS DESTINATION PORT:      443
INTERFACE INPUT:            Fa0/0
FLOW DIRECTION:             Input
FLOW SAMPLER ID:            0
IP PROTOCOL:                6
IP TOS:                     0x00
ip source as:               0
ip destination as:          0
ipv6 next hop address:      FE80::217:5AFF:FE4B:5821
ipv6 source mask:           /64
ipv6 destination mask:      /64
tcp flags:                  0x14
interface output:           Fa0/1
counter bytes:              60
counter packets:            1
timestamp first:            14:54:06.087
timestamp last:             14:54:06.087
```

Obrázek 4.39: *Výpis cache pro flow monitor IPv6 na směrovači AR6 - druhá část*

Pomocí příkazu **show IPv6 interface brief** na směrovačích AR5 a AR6 si lze vypsat přiřazení určité globální a lokální IPv6 adresy na rozhraní. Lokální IPv6 adresy se ve výpisech občas také vyskytují.

```
AR5#show ipv6 interface brief
FastEthernet0/0            [up/up]
    FE80::217:5AFF:FE4B:5820
    2001:DB8:1001:57::1
FastEthernet0/1            [up/up]
    FE80::217:5AFF:FE4B:5821
    2001:DB8:1001:58::1
```

Obrázek 4.40: *Výpis IPv6 adres přiřazených na síťových rozhraních na směrovači AR5*

```
AR6#show ipv6 interface brief
FastEthernet0/0            [up/up]
    FE80::21E:F7FF:FEAC:4A62
    2001:DB8:1001:59::1
FastEthernet0/1            [up/up]
    FE80::21E:F7FF:FEAC:4A63
    2001:DB8:1001:58::2
```

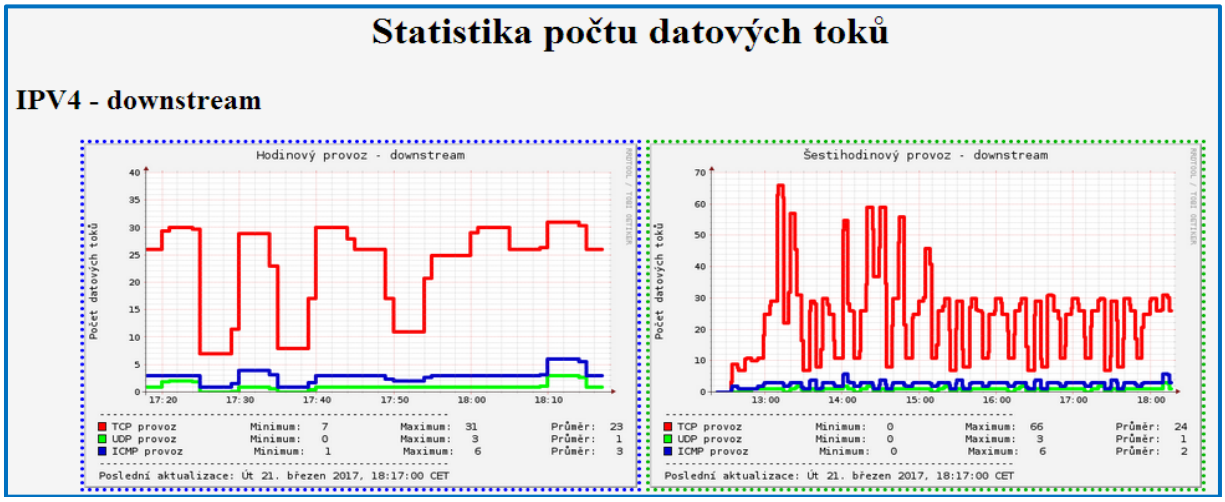
Obrázek 4.41: *Výpis IPv6 adres přiřazených na síťových rozhraních na směrovači AR6*

4.2.1. Výpisy z kolektoru (PC3-Collector)

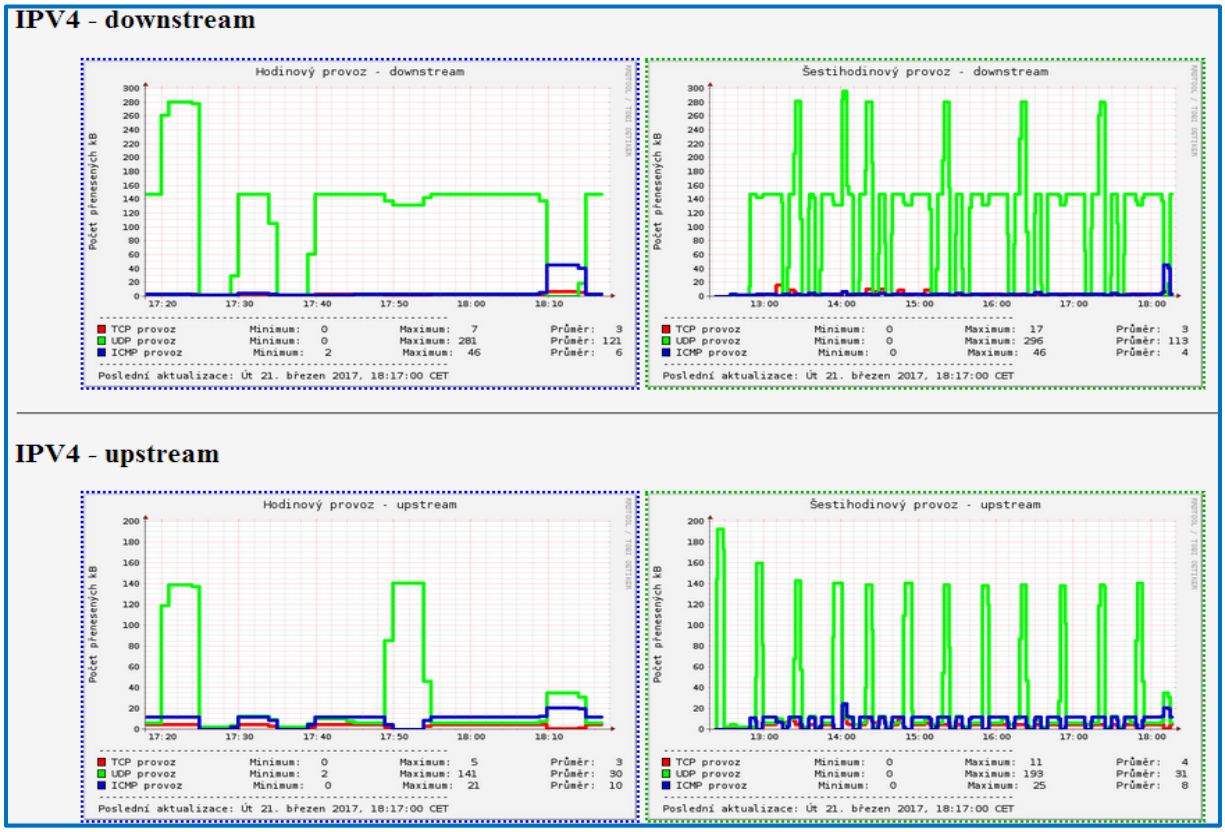
Přijatý provoz ze směrovačů AR5 a AR6 je podobný, jako v předchozí kapitole na počítači PC5_Collector. Proto není třeba znovu popisovat. Všechny soubory se zachyceným provozem jsou umístěny v příloze.

4.2.2. Grafy

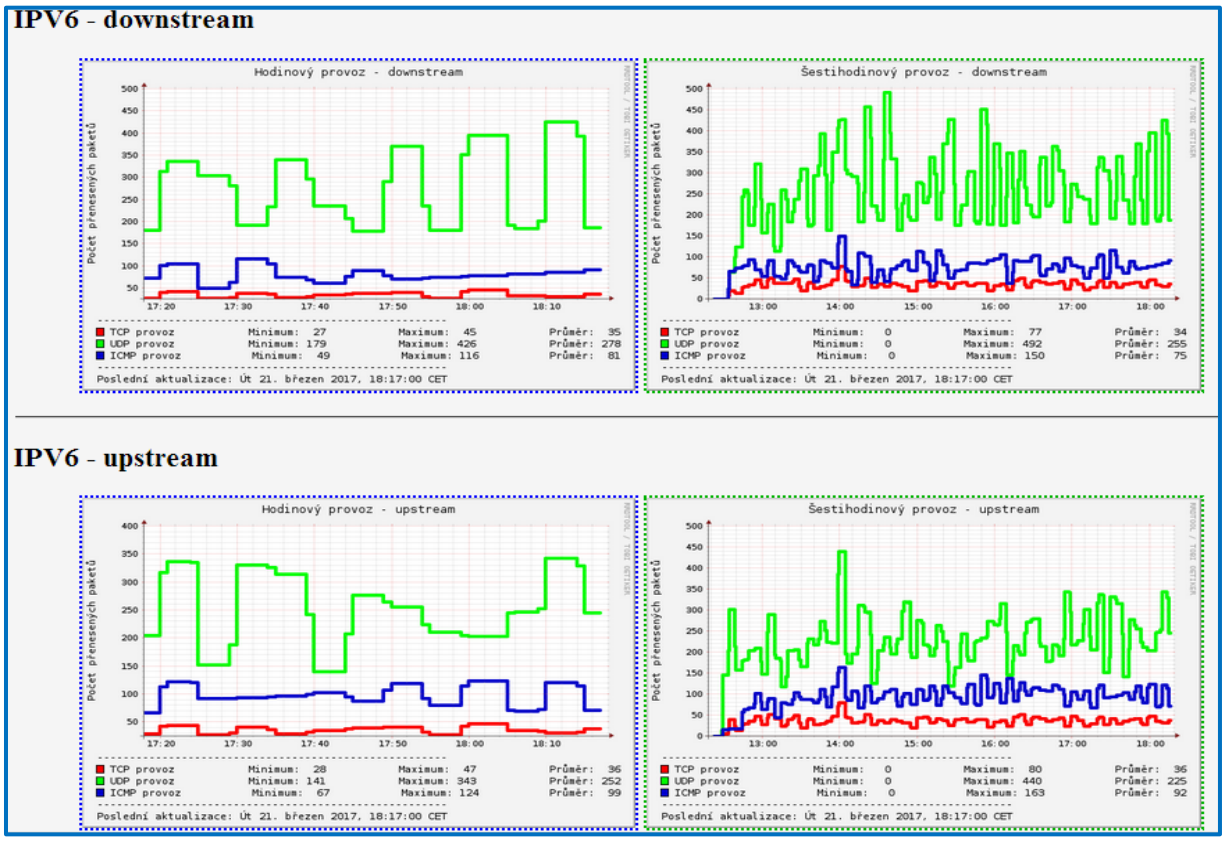
I zde bylo výstupem skriptu celkem 24 grafů pro počet přenesených kB, paketů a datových toků. Ke každému protokolu (IPv4, IPv6) a směru provozu (downstream, upstream) byl vždy vytvořen jeden hodinový a jeden šestihodinový graf. Grafy se zobrazovaly na webu. Nadpis první úrovně vždy popisoval, co tyto grafy monitorují. Níže jsou uvedeny některé z nich, všechny grafy jsou pak umístěny v příloze [16].



Obrázek 4.42: *Grafy zobrazující počet datových toků pro downstream (IPv4)*



Obrázek 4.43: *Grafy zobrazující počet přenesených kB (IPv4)*



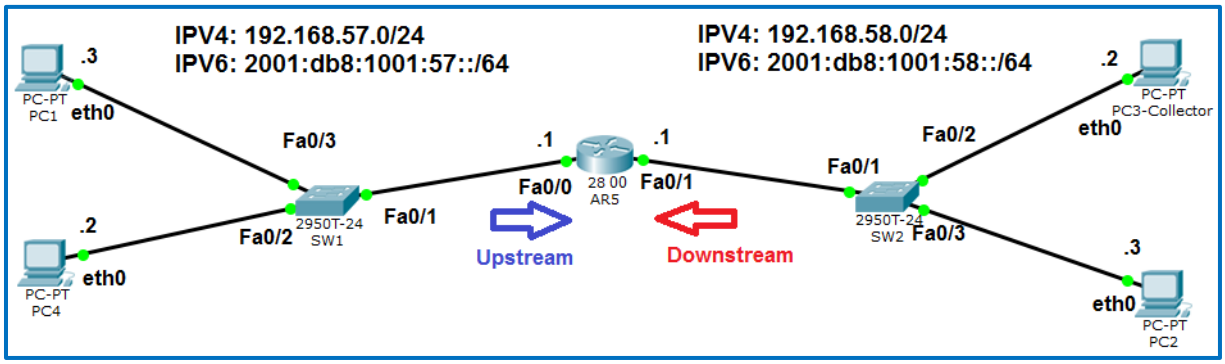
Obrázek 4.44: *Grafy zobrazující počet přenesených paketů (IPv6)*

4.3. Konfigurace Flexibilního NetFlow na Cisco 2800 pro IPv4 a IPv6

V poslední praktické části se budeme věnovat konfigurace Flexibilního NetFlow pro sběr IPv4 a IPv6 datových toků. Oproti předchozím konfiguracím, zde budou použity dva přepínače a jeden směrovač, který bude na vstupních rozhraních analyzovat provoz. Jak již bylo zmíněno v teoretické části - při použití Flexibilního NetFlow je možno vytvořit libovolný počet flow exportérů a flow monitorů, které lze pak přiřadit na určité rozhraní. Díky tomu jeden směrovač zvládne analyzovat provoz separátně pro všechna rozhraní. Oproti předcházející praktické části, zde již není použita defaultní šablona k analýze provozu, ale je zde nastaveno podle jakých parametrů se bude daný provoz analyzovat, a které parametry se budou odesílat na kolektor [14][15].

4.3.1. Topologie

Topologie se skládala ze dvou Cisco přepínačů a jednoho Cisco směrovače, na kterém bylo aktivováno Flexibilní NetFlow. K přepínači SW1 byly připojeny dva počítače (PC1 a PC4) a přepínači SW2 byly připojeny také dva počítače (PC2 a PC3_Collector, na kterém byl aktivován sběr NetFlow dat přijatých ze směrovače AR5) . Na směrovači AR5 byla použita verze IOSu 12.4(22)T, tedy stejná, jako v předchozí kapitole. Flexibilní NetFlow bylo aplikováno na vstupu rozhraní Fa0/0 (upstream) a Fa0/1 (downstream) pro sběr IPv4 a IPv6 datových toků. Topologie sítě s adresováním jednotlivých stanic je uvedena na obrázku níže.



Obrázek 4.45: *Topologie sítě*

4.3.2. Konfigurace na směrovači AR5

Nejdříve je nutno si pojmenovat směrovač (v našem případě AR3) a nakonfigurovat přeposílání IPv6 unicastových paketů. Aby bylo možno aktivovat sběr IPv4 a IPv6 provozu na rozhraních, je nutno také aktivovat CEF (Cisco Express Forwarding) pro IPv4 a IPv6.

```
Router(config)#hostname AR5
AR5(config)#IPv6 unicast-routing
AR5(config-if)#IP cef
AR5(config-if)#IPv6 cef
AR5#clock set 12:00:00 21 Mar 2017
```

Dále je nutno nakonfigurovat IPv4 a IPv6 adresy na jednotlivé rozhraní a aktivovat tato rozhraní. Konfigurace je uvedena pro obě rozhraní směrovače.

```
AR5(config)#interface FastEthernet 0/0
AR5(config-if)#ip address 192.168.57.1 255.255.255.0
AR5(config-if)#ip address 2001:db8:1001:57::1/64
AR5(config-if)#no shutdown
AR5(config)#interface FastEthernet 0/1
AR5(config-if)#ip address 192.168.58.1 255.255.255.0
AR5(config-if)#ip address 2001:db8:1001:58::1/64
AR5(config-if)#no shutdown
```

Po nakonfigurování adres aktivujeme směrovací algoritmus OSPF pro IPv4 a OSPVv3 pro IPv6 na jednotlivých rozhraních. Jako router -id ve směrovacím algoritmu OSPVv3 jsem zvolil 1.1.1.1.

```
AR5(config)#router ospf 1
AR5(config-router)#network 192.168.57.0 0.0.0.255 area 0
AR5(config-router)#network 192.168.58.0 0.0.0.255 area 0

AR5(config)#ipv6 router ospf 1
AR5(config-router)#router-id 1.1.1.1

AR5(config)#interface FastEthernet 0/0
AR5(config-if)#ipv6 ospf 1 area 0

AR5(config)#interface FastEthernet 0/1
AR5(config-if)#ipv6 ospf 1 area 0
```

Nyní zbývá nakonfigurovat Flexibilní NetFlow pro sběr IPv4 a IPv6 datových toků. Nejdříve bude popsána konfigurace pro sběr IPv4 datových toků.

Na začátku je nutné definovat, podle jakých kriterií se bude provoz analyzovat a které parametry datových toků se budou posílat na kolektor. K tomu slouží záznamy (records). Jako název byl zvolen **FLOW-RECORD-IPV4**. Na začátku definujeme pomocí klíčových polí (příkaz **match**), že chceme sbírat všechen provoz, který obsahuje IPv4 zdrojovou a cílovou adresu. Pak definujeme pomocí neklíčových polí (příkaz **collect**), které parametry budeme sbírat. V tomto případě budeme sbírat typ protokolu (IPv4 protokol), zdrojový port, cílový port, počet přenesených bytů a časové razítko začátku a konce datového toku. Oproti defaultní šabloně nesbíráme například počet přenesených paketů v datovém toku. Tato konfigurace se bude používat pro oba směry provozu protokolu IPv4.

```
AR5(config)# flow record FLOW-RECORD-IPV4
AR5(config-flow-record)# match ipv4 source address
AR5(config-flow-record)# match ipv4 destination address
AR5(config-flow-record)# collect ipv4 protocol
AR5(config-flow-record)# collect transport source-port
AR5(config-flow-record)# collect transport destination-port
AR5(config-flow-record)# collect counter bytes
AR5(config-flow-record)# collect timestamp sys-uptime first
AR5(config-flow-record)# collect timestamp sys-uptime last
```

Poté je třeba nakonfigurovat flow exportér pro oba směry provozu zvlášť. Název pro flow exporéry byl zvolen **FLOW-EXPORTER-IPv4-UP** pro upstream provoz a **FLOW-EXPORTER IPv4-DOWN** pro downstream provoz. Nejdříve nakonfigurujeme flow exportér pro upstream provoz, kde definujeme IP adresu rozhraní kolektoru - 192.168.58.2, zdrojové rozhraní - FastEthernet0/0 a UDP port 9954. Pro dowstream provoz je změna pouze ve zdrojovém rozhraní - FastEthernet0/1 a cílovém portu - UDP 9964.

```
AR5(config)#flow exporter FLOW-EXPORTER-IPV4-UP
AR5(config-flow-exporter)#destination 192.168.58.2
AR5(config-flow-exporter)#source FastEthernet0/0
AR5(config-flow-exporter)#transport udp 9954

AR5(config)#flow exporter FLOW-EXPORTER-IPV4-DOWN
AR5(config-flow-exporter)#destination 192.168.58.2
AR5(config-flow-exporter)#source FastEthernet0/1
AR5(config-flow-exporter)#transport udp 9964
```

Nakonec se dané flow exportéry a record definují ve flow monitorech a ty se nakonec přiřadí k danému rozhraní. Opět byly vytvořeny dva flow monitory - pro každý směr provozu jeden. Jako název pro jednotlivé flow monitory byl zvolen **FLOW-MONITOR-IPV4-UP** pro upstream, respektive **FLOW-MONITOR-IPV4-DOWN** pro downstream.

```
AR5(config)#flow monitor FLOW-MONITOR-IPV4-UP
AR5(config-flow-monitor)# record FLOW-RECORD-IPV4
AR5(config-flow-monitor)# exporter FLOW-EXPORTER-IPV4-UP

AR5(config)#flow monitor FLOW-MONITOR-IPV4-DOWN
AR5(config-flow-monitor)# record FLOW-RECORD-IPV4
AR5(config-flow-monitor)# exporter FLOW-EXPORTER-IPV4-DOWN

AR5(config)#interface FastEthernet0/0
AR5(config-if)#ip flow monitor FLOW-MONITOR-IPV4-UP input

AR5(config)#interface FastEthernet0/1
AR5(config-if)#ip flow monitor FLOW-MONITOR-IPV4-DOWN input
```

Nyní bude popsána konfigurace pro analýzu IPv6 provozu.

Opět je třeba definovat, podle jakých kriterií se bude provoz analyzovat a které parametry provozu se budou posílat na kolektor. Jako název pro flow record byl zvolen **FLOW-RECORD-IPV6**. Zde budeme analyzovat všechen provoz, který obsahuje IPv6 cílovou adresu. Pak definujeme pomocí neklíčových polí (příkaz **collect**), které parametry budeme sbírat. V tomto případě budeme sbírat typ protokolu (IPv6 protokol), IPv6 adresu (není uveden v klíčových polích, tudíž je třeba jí definovat zde), zdrojový port, počet přenesených bytů a časové razítko začátku a konce datového toku.

```
AR5(config)# flow record FLOW-RECORD-IPV4
AR5(config-flow-record)# match ipv6 destination address
AR5(config-flow-record)# collect ipv6 protocol
```

```
AR5(config-flow-record)# collect ipv6 source address
AR5(config-flow-record)# collect transport source-port
AR5(config-flow-record)# collect counter bytes
AR5(config-flow-record)# collect timestamp sys-uptime first
AR5(config-flow-record)# collect timestamp sys-uptime last
```

Poté je třeba nakonfigurovat flow exportér pro oba směry provozu zvlášť. Název pro flow exporéry byl zvolen **FLOW-EXPORTER-IPV6-UP** pro upstream provoz a **FLOW-EXPORTER IPV6-DOWN** pro downstream provoz. Nejdříve nakonfigurujeme flow exportér pro upstream provoz, kde definujeme IP adresu rozhraní kolektoru - 192.168.58.2, zdrojové rozhraní - FastEthernet0/0 a UDP port 9956. Pro dowstream provoz je změna pouze ve zdrojovém rozhraní - FastEthernet0/1 a cílovém portu - UDP 9966.

```
AR5(config)#flow exporter FLOW-EXPORTER-IPV6-UP
AR5(config-flow-exporter)#destination 192.168.58.2
AR5(config-flow-exporter)#source FastEthernet0/0
AR5(config-flow-exporter)#transport udp 9956

AR5(config)#flow exporter FLOW-EXPORTER-IPV6-DOWN
AR5(config-flow-exporter)#destination 192.168.58.2
AR5(config-flow-exporter)#source FastEthernet0/1
AR5(config-flow-exporter)#transport udp 9966
```

Nakonec se dané flow exportéry a record definují ve flow monitorech a ty se nakonec přiřadí k danému rozhraní. Opět byly vytvořeny dva flow monitory - pro každý směr provozu jeden. Jako název pro jednotlivé flow monitory byl zvolen **FLOW-MONITOR-IPV6-UP** pro upstream, respektive **FLOW-MONITOR-IPV6-DOWN** pro downstream.

```
AR5(config)#flow monitor FLOW-MONITOR-IPV6-UP
AR5(config-flow-monitor)# record FLOW-RECORD-IPV6
AR5(config-flow-monitor)# exporter FLOW-EXPORTER-IPV6-UP

AR5(config)#flow monitor FLOW-MONITOR-IPV6-DOWN
AR5(config-flow-monitor)# record FLOW-RECORD-IPV6
AR5(config-flow-monitor)# exporter FLOW-EXPORTER-IPV6-DOWN

AR5(config)#interface FastEthernet0/0
AR5(config-if)#ipv6 flow monitor FLOW-MONITOR-IPV6-UP input

AR5(config)#interface FastEthernet0/1
AR5(config-if)#ipv6 flow monitor FLOW-MONITOR-IPV6-DOWN input
```

4.3.1. Výpisy tabulek ze směrovačů

Následující výpis ze směrovače AR5 ukazuje výpis cache pro IPv4 protokol a směr downstream (**FLOW-MONITOR-IPV4-DOWN**). Na začátku si lze přečíst, že cache má kapacitu 4096 záznamů a aktuálně jsou zde uloženy 4 záznamy o datových tocích. Dále je zde vypsát jeden ze záznamů, kde je vidět, že obsahuje přesně takové pole, které jsme si definovali v sekci flow record. Například oproti

defaultnímu nastavení, toto nastavení neobsahuje počet paketů. Lze si rovněž všimnout, že klíčová pole v záznamu jsou vyznačeny velkými písmeny (**IPV4 SOURCE ADDRESS, IPV4 DESTINATION ADDRESS**). Je tady i menší odlišnost ve značení použitého typu protokolu (pole "**ip protocol**") - není zde uvedeno klasické označení protokolu TCP, UDP nebo ICMP, ale používají se zde čísla. Například v tomto případě číslo "6" patří k TCP protokolu.

```
AR5#show flow monitor FLOW-MONITOR-IPV4-DOWN cache
  Cache type:                               Normal
  Cache size:                                 4096
  Current entries:                               4
  High Watermark:                                4

  Flows added:                                  52
  Flows not added:                               5
  Flows aged:                                   48
    - Active timeout    (  1800 secs)            1
    - Inactive timeout (    15 secs)            47
    - Event aged                                 0
    - Watermark aged                             0
    - Emergency aged                             0
```

Obrázek 4.46: *Výpis cache pro flow monitor FLOW-MONITOR-IPv4-DOWN na směrovači AR5- první část*

```
IPV4 SOURCE ADDRESS:       192.168.58.3
IPV4 DESTINATION ADDRESS:  192.168.57.3
trns source port:          80
trns destination port:     80
counter bytes:             480
timestamp first:           16:29:41.855
timestamp last:            16:30:12.147
ip protocol:               6
```

Obrázek 4.47: *Výpis jednoho z více záznamů z cache pro flow monitor FLOW-MONITOR-IPv4-DOWN na směrovači AR5- druhá část*

Výpis cache pro směr upstream (**FLOW-MONITOR-IPv4-UP**) pro protokol IPv4 je podobný, liší se prefixy ve zdrojové adrese a cílové adrese.

Na následujících obrázcích 4.48, 4.49 lze vidět výpis cache pro IPv6 protokol a směr upstream (**FLOW-MONITOR-IPV6-UP).** Struktura cache je stejná, jako u předchozího výpisu. Opět je zde vypsán jeden ze záznamů, který se aktuálně nachází v cache. Informace o datových tocích v téhle cache byly umístěny na základě cílové IPv6 adresy. Navíc oproti předchozímu výpisu, zde chybí ještě navíc cílový port, který záměrně nebyl definován v sekci flow record. Opět se zde objevují IPv6 adresy, které byly automaticky přiřazeny na síťové rozhraní počítačů.

```
AR5#show flow monitor FLOW-MONITOR-IPV6-UP cache
  Cache type:                               Normal
  Cache size:                                 4096
  Current entries:                               3
  High Watermark:                                6

  Flows added:                                 218
  Flows not added:                               1
  Flows aged:                                  215
    - Active timeout   (  1800 secs)             0
    - Inactive timeout (    15 secs)           215
    - Event aged                                 0
    - Watermark aged                             0
    - Emergency aged                             0
```

Obrázek 4.48: *Výpis cache pro flow monitor FLOW-MONITOR-IPv6-UP na směrovači AR5- první část*

```
IPV6 DESTINATION ADDRESS:  2001:DB8:1001:58::3
ipv6 source address:       2001:DB8:1001:57:76D4:35FF:FE7C:29E6
trns source port:          443
counter bytes:             17232
timestamp first:           16:33:05.915
timestamp last:            16:33:16.915
ip protocol:               6
```

Obrázek 4.49: *Výpis jednoho z více záznamů z cache pro flow monitor FLOW-MONITOR-IPv6-UP na směrovači AR5- druhá část*

```
root@PC4:/home/student# ifconfig
eth0      Link encap:Ethernet  HWaddr 74:d4:35:7c:29:e6
          inet addr:192.168.57.2  Bcast:0.0.0.0  Mask:255.255.255.0
          inet6 addr: 2001:db8:1001:57::2/64 Scope:Global
          inet6 addr: fe80::76d4:35ff:fe7c:29e6/64 Scope:Link
          inet6 addr: 2001:db8:1001:57:76d4:35ff:fe7c:29e6/64 Scope:Global
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:33654 errors:0 dropped:57 overruns:0 frame:0
          TX packets:21982 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:14062772 (14.0 MB)  TX bytes:7463522 (7.4 MB)
```

Obrázek 4.50: *Výpis nastavení síťových rozhraní na PC4*

Lze si rovněž vypsal infomace o konfiguraci flow monitorů, exportéru nebo record. Příkazem **show flow monitor** si vypíšeme informace o všech nakonfigurovaných flow monitorem. Je zde například informace, který flow exportér nebo flow record je u tohoto flow monitoru aktivován. Pro ukázku zde uvedu výpis flow monitoru pro analýzu IPv6 datových toků ve směru upstream.

```
AR5#show flow monitor
Flow Monitor FLOW-MONITOR-IPV6-UP:
  Description:       User defined
  Flow Record:       FLOW-RECORD-IPV6
  Flow Exporter:     FLOW-EXPORTER-IPV6-UP
  Cache:
    Type:              normal
    Status:            allocated
    Size:              4096 entries / 311316 bytes
    Inactive Timeout:  15 secs
    Active Timeout:    1800 secs
    Update Timeout:    1800 secs
```

Obrázek 4.51*: Výpis nastavení exportu flow monitoru FLOW-MONITOR-IPV6-UP*

4.3.1. Výpisy z kolektoru (PC3-Collector)

Stejně jako v předchozích částech, i zde jsou každých 5 minut vytvářeny soubory se zachyceným provozem. Tyto soubory později zpracovává skript **netflow_diplomka.sh** a ze zpracovaných hodnot vytváří grafy. Některý ze souborů si lze také vypsat manuálně pomocí příkazu **nfdump** (**nfdump -r nfcapd.201703291900,** soubor z 29.3. 2017). Následující výpis na kolektoru (PC3-Collector) ukazuje informace o datových tocích, které byly vytvořeny mezi 18:55 a 19:00. Jedná se o IPv4 datové toky ve směru upstream - to lze také poznat podle prefixů zdrojových a cílových IP adres.

Oproti předchozím výpisům si lze všimnout, že se zde u žádného datového toku nepočítá počet přenesených paketů (na směrovači AR5 nebyl aktivován v sekci flow record sběr informací o paketech pro IPv4 protokol).

```
Date flow start          Duration Proto      Src IP Addr:Port          Dst IP Addr:Port   Packets    Bytes Flows
Skip unknown record type 7
PANIC! - Verify map id 0: ERROR: element id 27 out of range [25]!
2017-03-29 18:55:17.320     0.000 TCP       192.168.57.3:81    ->     192.168.58.3:81          0       60     1
2017-03-29 18:56:00.592    16.164 TCP       192.168.57.2:23    ->     192.168.58.3:21          0     1628     1
2017-03-29 18:56:00.596    28.852 TCP       192.168.57.3:23    ->     192.168.58.3:23          0     8328     1
2017-03-29 18:56:32.484     0.000 UDP       192.168.57.3:0     ->     192.168.59.2:20          0      140     1
2017-03-29 18:56:32.576     0.384 UDP       192.168.57.3:30000 ->     192.168.58.2:30000       0     4000     1
2017-03-29 18:57:00.776     0.000 TCP       192.168.57.2:23    ->     192.168.58.2:23          0       60     1
2017-03-29 18:57:01.364    11.000 TCP       192.168.57.3:21    ->     192.168.58.3:23          0      120     1
2017-03-29 18:57:13.784     2.996 UDP       192.168.57.2:53    ->     192.168.58.3:53          0     6300     1
2017-03-29 18:57:14.368     1.800 UDP       192.168.57.3:53    ->     192.168.58.2:53          0     2840     1
2017-03-29 18:58:13.800     1.800 UDP       192.168.57.2:53    ->     192.168.58.2:53          0     2840     1
Summary: total flows: 10, total bytes: 26316, total packets: 0, avg bps: 1180, avg pps: 0, avg bpp: 0
Time window: 2017-03-29 18:55:17 - 2017-03-29 18:58:16
Total flows processed: 11, Blocks skipped: 0, Bytes read: 808
Sys: 0.000s flows/second: 0.0        Wall: 0.001s flows/second: 8154.2
```

Obrázek 4.52: *Výpis IPv4 datových toků na kolektoru (PC3_Collector)*

Pro IPv6 datové toky je výpis podobný. Zde se ovšem vyskytují IPv6 adresy. Lze si všimnout, že se zde vyskytují i jiné IPv6 adresy se stejným prefixem, jako byly nakonfigurovány na jednotlivé počítače. Jedná se o globální IPv6 adresy, které byly na jednotlivá rozhraní přidány automaticky. Přiřazení jednotlivých IPv6 adres si lze zobrazit pomocí příkazu **ifconfig**.

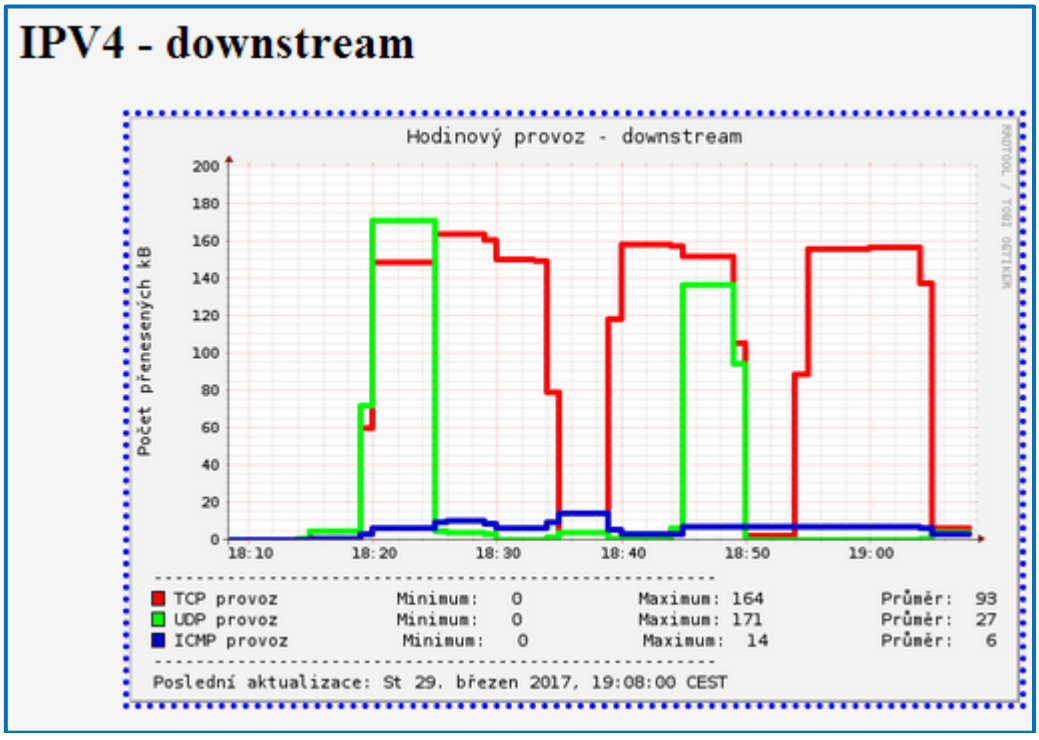
Opět oproti předchozím výpisům na kolektorech, zde se také pro žádný datový tok nepočítá jeho počet paketů ani nebyly analyzovány cílové porty, místo toho je zde číslo 0.

```
Date flow start          Duration Proto      Src IP Addr:Port          Dst IP Addr:Port   Packets    Bytes Flows
Skip unknown record type 7
PANIC! - Verify map id 0: ERROR: element id 27 out of range [25]!
2017-03-29 18:55:08.376    10.008 ICMP6 fe80::7..7c:29f0.0     -> fe80::2..4b:5821.0.0         0      400     1
2017-03-29 18:55:03.372    18.536 UDP   2001:db..7c:29f0.43920 -> 2001:71..001::53.0           0     1312     1
2017-03-29 18:56:05.596     6.684 ICMP6 2001:db..7c:29f0.0     -> fe80::2..4b:5821.0.0         0      392     1
2017-03-29 18:56:00.756    13.000 TCP   2001:db..1:58::2.5000  -> 2001:db..7c:29e6.0           0     5044     1
2017-03-29 18:56:13.596     0.000 TCP   2001:db..7c:29f0.20    -> 2001:db..1:57::3.0           0       80     1
2017-03-29 18:56:00.592    25.004 TCP   2001:db..7c:29f0.80    -> 2001:db..1:57::2.0           0    12336     1
2017-03-29 18:56:30.596     0.000 TCP   2001:db..7c:29f0.32323 -> 2001:db..1:57::3.0           0       80     1
2017-03-29 18:56:00.596    35.036 UDP   2001:db..7c:29f0.48876 -> 2001:71..001::53.0           0     2296     1
2017-03-29 18:56:30.756     9.092 ICMP6 2001:db..1:58::2.0     -> 2001:db..7c:29e6.0.0         0     9120     1
2017-03-29 18:56:01.344    40.864 TCP   2001:db..1:58::2.5000  -> 2001:db..7c:2d7d.0           0    33073     1
Summary: total flows: 10, total bytes: 64133, total packets: 0, avg bps: 5191, avg pps: 0, avg bpp: 0
Time window: 2017-03-29 18:55:03 - 2017-03-29 18:59:28
Total flows processed: 19, Blocks skipped: 0, Bytes read: 1808
Sys: 0.000s flows/second: 0.0        Wall: 0.001s flows/second: 14694.5
```

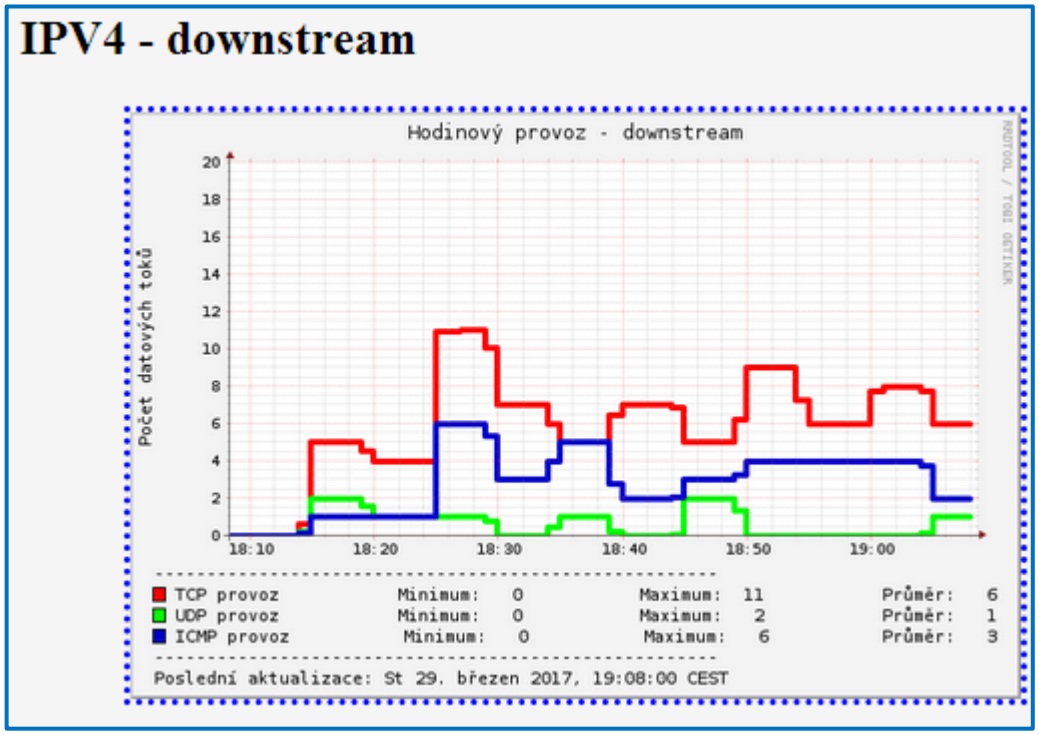
Obrázek 4.53: *Výpis IPv6 datových toků na kolektoru (PC3_Collector)*

4.3.2. Grafy

Výstupem skriptu bylo celkem 24 grafů pro počet přenesených kB, paketů a datových toků. Avšak vzhledem k tomu, že počet přenesených paketů pro každý datový tok se nepočítal, zůstaly hodnoty v grafech na hodnotě 0. Grafy se zobrazovaly opět na webové stránce, která se automaticky po minutě aktualizovala, aby zobrazovala aktuální obrázky grafů [16].



Obrázek 4.54: *Graf zobrazující počet přenesených kB pro downstream (IPv4)*



Obrázek 4.55: *Graf zobrazující počet datových toků pro downstream (IPv4)*

5. Závěr

Cílem diplomové práce bylo navrhnout řešení pro monitorování síťového provozu pomocí nástroje NetFlow, a jeho následné vykreslování pomocí nástroje RRDTool.

Výstupem této práce je popis protokolu NetFlow, jeho funkčnosti a schémat zapojení. Kromě protokolu NetFlow je zde popsán i jeho nástupce na Cisco zařízeních - Flexibilní NetFlow. Dále je v teoretické části popsán nástroj RRDTool a různé open-source NetFlow analyzátory, především analyzátory Softflowd a Nfdump.

Dále je zde popsána konfigurace NetFlow na počítačích s operačním systémem Linux, kde jeden počítač (NetFlow exportér) analyzuje provoz na jeho síťovém rozhraní a druhý počítač (NetFlow kolektor) tyto data zpracovává a vytváří z nich grafy pomocí nástroje RRDTool.

V následující části je popis konfigurace NetFlow protokolu na Cisco zařízeních (směrovačích) s různými verzemi IOSu. Topologie se skládá ze dvou Cisco směrovačů, které analyzují provoz - každý v obou směrech a výsledná NetFlow data posílají na kolektor, kterým je jeden z počítačů v síti. Tento počítač následně data zpracuje a vytvoří celkem 24 grafů, ve kterých je znázorněn počet přenesených kB, paketů a toků pro IPv4 a IPv6 protokol. Rovněž je zde popsána konfigurace Flexibilního NetFlow, včetně detailních výpisů.

Mimo práci jsem se také zabýval možností konfigurace NetFlow na Cisco přepínačích. Jednalo se převážně o Cisco přepínače 3560 X-series, kde se zdálo, že by mohl podporovat i Flexibilní NetFlow. Nicméně u tohoto přepínače nebylo možno aplikovat Flexibilní NetFlow na kterékoliv síťové rozhraní. U klasického NetFlow podporoval tento Cisco přepínač pouze protokol IPv4, avšak ani po správné konfiguraci žádné datové toky neanalyzoval ani neposílal na kolektor. Podle oficiálních webových stránek od firmy Cisco by měl tento protokol fungovat na vyšších řadách Cisco přepínačů (Cisco 3750 X-series).

Z pohledu dalšího vývoje práce by bylo vhodné otestovat v topologii také NetFlow sondy, které byly popsány v kapitole 1. Rovněž by bylo vhodné vyzkoušet protokol NetFlow na novějších Cisco přepínačích, které by tento protokol podporovaly. Co se týče vyhodnování výsledků a vytváření grafů - nebylo by na škodu vyzkoušet i jiný program na vykreslování grafů. V případě nástroje RRDTool jsem narážel na některé jeho nedostatky - barvu pod křivkou v grafu dokázal vykreslit maximálně pro 2 proměnné, nebo po skokovém zvýšení některé z hodnot tuto hodnotu nezapsal do grafu s dostatečnou přesností (odchylka oproti skutečné hodnotě byla i 10 %). Kromě grafů by bylo možno použít i tabulky, ve kterých by se vypisovaly třeba zdrojové IP adresy s největším počtem datových toků nebo přenesených dat, díky čemuž by síťový administrátor mohl snadno odhalit případný DoS útok.

Použitá literatura

[1] GALČAN, Daniel. *Analýza IP toků pomocí OpenSource utilit, implementace NetFlow* [online]. 2009, [cit. 2017-04-05]. Dostupné z: <http://wh.cs.vsb.cz/sps/images/f/f6/Galcan-NetFlow-opensource.pdf>

[2] *What is NetFlow?* [online]. 2006 [cit. 2017-04-05]. Dostupné z: <http://NetFlow.caligare.com/>.

[3] TOMAN, Adrian; STOSZEK, Jakub. *Možnosti sběru informací ze směrovačů Cisco pomocí NetFlow* [online]. 2008 [cit. 2017-04-05]. Dostupné z: < http://wh.cs.vsb.cz/sps/images/8/89/NetFlow.pdf >

[4] *Flexible NetFlow* [online]. 2016, [cit. 2017-04-05]. Dostupné z: <http://www.cisco.com/c/en/us/products/ios-nx-os-software/flexible-NetFlow/index.html>.

[5] *Cisco IOS Flexible NetFlow Overview* [online]. 2011, [cit. 2017-04-05]. Dostupné z: <http://www.cisco.com/c/en/us/td/docs/ios/fnetflow/configuration/guide/12_2sr/fnf_12_2_sr_book/cust_fnflow_rec_mon.html >

[6] *Nfdump* [online]. 2014, [cit. 2017-04-05]. Dostupné z: < http://nfdump.sourceforge.net/>

[7] *SoftFlowd* [online]. 2011, [cit. 2017-04-05]. Dostupné z: < http://www.mindrot.org/projects/softflowd/>

[8] *SoftFlowd - Traffic flow monitoring* [online]. 2011, [cit. 2017-04-05]. Dostupné z: < http://manpages.ubuntu.com/manpages/precise/man8/softflowd.8.html>

[9] *Free open source analyzers for Windows and Unix/Linux* [online]. [cit. 2017-04-05]. Dostupné z: <http://www.pcwdld.com/free-open-source-NetFlow-analyzers> .

[10] DOLSON, Van, Ray. *Top 5 open source network analyzers* [online]. [cit. 2017-04-05]. 2013. Dostupné z: <http://techteapot.com/top-5-open-source-NetFlow-analyzers/>

[11] *RRDTool* [online]. [cit. 2017-04-05]. 2016. Dostupné z: < https://cs.wikIPedia.org/wiki/RRDTool >

[12] *Configuring NetFlow and NetFlow data export* [online]. [cit. 2017-04-05]. 2011. Dostupné z: < http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/NetFlow/configuration/xe-3s/nf-xe-3s-book/cfg-nflow-data-expt-xe.html>

[13] *Implementing NetFlow for IPv6* [online]. [cit. 2017-04-05]. 2011. Dostupné z: <http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/IPv6/configuration/12-2sx/IPv6-12-2sx-book/IP6-NetFlow.html>

[14] SIVAGAMI NARAYANAN, *IPv6 Flexible NetFlow Configuration Example* [online]. [cit. 2017-04-05]. 2012. Dostupné z: < https://supportforums.cisco.com/document/105221/IPv6-flexible-NetFlow-configuration-example>

[15] *Configuring Flexible NetFlow* [online]. [cit. 2017-04-05]. 2013. Dostupné z: <http://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst3750x_3560x/software/release/15-2_2_e/flexible_netflow/configuration_guide/b_fnf_1522e_3750x_3560x_cg/b_fnf_1522e_3750x_3560x_cg_chapter_010.html>

[16] JAN KUS, *Hrátky s RRDTool* [online]. [cit. 2017-04-05]. 2008. Dostupné z: < https://jankus.cz/?id=3&t=127>

[17] *MGEN User's and Reference Guide Version 5.0* [online]. [cit. 2017-04-05]. 2009. Dostupné z: <https://downloads.pf.itd.nrl.navy.mil/docs/mgen/mgen.html>

[18] *MausezahnUser's Guide* [online]. [cit. 2017-04-05]. 2010. Dostupné z: <http://www.perihel.at/sec/mz/mzguide.html#udp>

[19] *Testing firewall rules with Hping3 - examples* [online]. [cit. 2017-04-05]. 2010. Dostupné z: <http://0daysecurity.com/articles/hping3_examples.html>

[20] *NetFlow, nová éra monitorování počítačových sítí* [online]. [cit. 2017-04-05]. 2015. Dostupné z: <https://www.flowmon.com/cs/solutions/use-case/netflow-IPfix>

Seznam příloh